DECOMPOSITION AND STOCHASTIC OPTIMIZATION METHODS FOR MACHINE LEARNING

Hongsheng Liu

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2020

Approved by:

Shu Lu

Amarjit Budhiraja

Yufeng Liu

Dzung Phan

Quoc Tran-Dinh

ABSTRACT

HONGSHENG LIU: DECOMPOSITION AND STOCHASTIC OPTIMIZATION METHODS FOR MACHINE LEARNING.
(Under the direction of Shu Lu.)

Machine learning has been a topic in academia and industry for decades. Performance of machine learning heavily relies on efficiency of its underlying algorithms. Optimization is one of the core components in machine learning algorithms. This thesis focuses on decomposition and subsampled optimization methods that can be used in various machine learning problems.

The first two chapters aim to solve the constrained optimization problems with block structures. In Chapter 2, we propose the ADA algorithm which optimizes in parallel. The global convergence and local convergence rate are established. The inexact version of the ADA is introduced and studied. In Chapter 3, we develop the two-level ADMM which can remedy the divergence of multi-block ADMM. Both theoretical convergence and numerical experiments show the advantages of the new algorithms over classical methods.

In Chapter 4, We consider the composite convex minimization problem of a finite sum and a nonsmooth convex regularizer which covers various machine learning and statistics applications. We develop a novel subsampled proximal Newton method under the *generalized self-concordant* assumption on the loss function.

In the final chapter, we focus on the hyperparameter tuning optimization for machine learning. Based on the ideas from the well-known Bayesian optimization and DIRECT algorithms, we propose a hybrid Bayesian optimization method which provides a new strategy to trade-off between exploitation and exploration.

ACKNOWLEDGMENTS

It's my great pleasure to work as a PhD student under the supervision of Prof. Shu Lu in the past five years. Her devotion, perseverance and insight for the research in optimization have deeply influenced me. She is also open-minded in research areas and gives me enough flexibility to explore possible research projects in optimization, machine learning and statistics. Her guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Amarjit Budhiraja, Prof. Yufeng Liu, Prof. Quoc Tran-Dinh and Dr. Dzung Phan, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

My sincere thanks also go to Dr. Dzung Phan, and Dr. Lam M. Nguyen from IBM, who provided me an opportunity to join their team as a intern, and who gave access to the laboratory and research facilities. Without they precious support it would not be possible to conduct this research.

I thank my fellow colleagues, Prof. Zhengling Qi, Dr. Yifan Cui, Dr. Tianxiao Sun and Dr. Jingxing Wang for the stimulating discussions, and for all the fun we have had in the past five years.

Finally, I would like to thank my parents and my wife for supporting me spiritually throughout writing this thesis.

TABLE OF CONTENTS

CHAPTER 1: Introduction

Machine learning is the study of computer algorithms that improve automatically through experience. With wide applications in daily life, machine learning has been a hot research topic among the academia society and the industries. The performance of machine learning algorithms heavily relies on the underlying optimization algorithms. Mathematical optimization serves as the engine for machine learning and receives more attentions in research.

This thesis focuses on a variety of optimization algorithms designed for machine learning problems. It covers three different topics: optimization for multi-block problems, Newton method for *generalized self-concordant* functions, and black-box optimization for hyperparameter tuning problems.

1.1 Optimization for multi-block problems

We begin with the constrained optimization problems with the following form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^{K} f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{K} A_i x_i = b \\ & x_i \in \chi_i, i = 1, \ldots, K. \end{aligned} \tag{1.1}$$

with $\mathbf{x} := (x_1, \ldots, x_K) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_K} = \mathbb{R}^n$, $\mathbf{A} := (A_1, \ldots, A_K) \in \mathbb{R}^{m \times n}$ and $\chi_i, i = 1, \ldots, K$ being simple convex sets. The objective functions f_i are not necessary convex or smooth.

Multi-block optimization problems arise in many areas such as signal processing, statistics and machine learning. The authors in [65] summarizes a list of applications arising from many areas. Many numerical algorithms have been proposed to solve them; see [15, 16, 18, 27, 69, 89, 101] and references therein. Among these methods, the ADMM method is perhaps the most popular

approach due to its suitable parallel implementation and outstanding computational performance.

The alternating direction method of multipliers (ADMM) [27] is a popular algorithm that solves convex optimization problems by breaking them into smaller pieces, each of which is then easier to handle. The algorithm originally solves (1.1) with $K = 2$ and convex target functions f_1 and f_2. As in the method of multipliers, we form the augmented Lagrangian

$$\mathcal{L}_\beta(x_1, x_2, y) = f_1(x_1) + f_2(x_2) + y^T(A_1x_1 + A_2x_2 - b) + \beta/2 \left\|A_1x_1 + A_2x_2 - b\right\|^2. \tag{1.2}$$

The ADMM is given as follows:

Algorithm 1.1 Alternating direction method of multipliers

1: Given $x_1^0, x_2^0, y^0, k \leftarrow 0$.
2: **while** not converged **do**
3: $\quad x_1^{k+1} := \operatorname{argmin}_{x_1 \in \chi_1} \mathcal{L}_\beta(x_1, x_2^k, y^k)$
4: $\quad x_2^{k+1} := \operatorname{argmin}_{x_2 \in \chi_2} \mathcal{L}_\beta(x_1^{k+1}, x_2, y^k)$
5: $\quad y^{k+1} := y^k + \beta(A_1x_1^{k+1} + A_2x_2^{k+1} - b)$
6: $\quad k \leftarrow k + 1$
7: Output x_1^k, x_2^k, y^k.

It has recently found wide application in the form of (1.1) in a number of areas, including statistics, machine learning, computer science and engineering. Numerous work has been done in terms of both the theoretical convergence results and empirical applications [11, 15, 36, 37, 42, 103]. The success of applying Algorithm 1.1 in the convex and two-block decomposition problem has inspired researcher to extend ADMM to a more general setting. We let $x_{<i} := [x_1; \ldots; x_{i-1}] \in \mathbb{R}^{n_1+\cdots+n_{i-1}}$ and $x_{>i} := [x_{i+1}; \ldots; x_K] \in \mathbb{R}^{n_{i+1}+\cdots+n_K}$ (clearly, $x_{<1}$ and $x_{>K}$ are null variables, which may be used for notational ease). Subvectors $x_{\leq i} := [x_{<i}, x_i]$ and $x_{\geq i}$ are defined similarly. Similarly, we can define the augmented Lagrangian function

$$\mathcal{L}_\beta(\mathbf{x}, y) = \sum_{i=1}^{K} f_i(x_i) + \left\langle y, \sum_{i=1}^{K} A_ix_i - b \right\rangle + \frac{\beta}{2} \left\| \sum_{i=1}^{K} A_ix_i - b \right\|^2. \tag{1.3}$$

The direct extension of ADMM to the multi-block problem is given in Algorithm 1.2. However, it has been shown in [17] that the above extension may not converge even in the convex case when $K > 2$. In many applications, the target functions f_i are also not necessary convex which poses

Algorithm 1.2 Multi-block ADMM

Given $\mathbf{x}^0, y^0, k \leftarrow 0$.
while not converged **do**
  for $i = 1, \ldots, K$ **do**
    $x_i^{k+1} \leftarrow \text{argmin}_{x_i \in \chi_i} \mathcal{L}_\beta(x_{<i}^{k+1}, x_i, x_{>i}^k, y^k)$;
  $y^{k+1} \leftarrow y^k + \beta(\mathbf{A}\mathbf{x}^{k+1} - b)$;
  $k \leftarrow k + 1$;
Output $\mathbf{x}^k, y^k$.

another challenge for the applications of Algorithm 1.2.

We summarize the our contributions in this topic as follows. In Chapter 2, we propose the Augmented Decomposition Algorithm (ADA) to solve problem (1.1) in parallel. Different from the sequential update in ADMM, ADA decomposes the original problem into K blocks and solves them individually. The dual step collects information from each local solver and updates the dual parameters in parallel. Numerical examples in machine learning problems illustrate the stability and efficiency of ADA.

In Chapter 3, in order to resolve the difficulty of ADMM in multi-block problems, we develop a two-level ADMM algorithm with theoretical guarantee. It relaxes two crucial assumptions for the classical ADMM and achieves fast convergence results in practical applications.

1.2 Stochastic Newton method for empirical risk minimization problem

Unconstrained optimization problems receive even more attention in the machine learning community as more models belong to this category. Many machine learning algorithms aim to solve the empirical risk minimization problem

$$\min_{w \in \mathbb{R}^p} \left\{ F(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w) + g(w) \right\} \tag{1.4}$$

where $f_i(\cdot)$ represents the loss induced by each data point and $g(\cdot)$ is the regularization term which prevents model overfitting. Due to the large sample size n, researchers are more interested in the stochastic method. The stochastic gradient descent (SGD) [9] has become a standard tool to optimize (1.4). However, SGD suffers heavily from the high variance and cannot produce high quality solutions. Recently, many variance reduction methods have been proposed to mitigate the

drawbacks of SGD, including SVRG [109], SDCA [87], SARAH [73], and SAGA [22]. All of these algorithms relies on the first order information $\nabla f_i(w)$ and the convergence results are based on the assumption of Lipschitz continuous gradient of f_i.

We start from a different point view of (1.4) and make use of the second order information of f_i. Unlike the first order methods, the second order method can achieve highly accurate solutions and converge quadratically in the local region. However, each iteration of the second order method is prohibitively expensive due to the evaluation of the Hessian matrix. On the other hand, line-search is often used to select the learning rate. It is unrealistic for the empirical risk minimization problems as each function evaluation needs to loop all data points.

In Chapter 4, we propose a stochastic newton method under the *generalized self-concordant* assumption on the loss function f_i. By utilizing the special property of *generalized self-concordant* functions, the new algorithm is able to determine an optimal learning rate in the global optimization steps. In the local region, the new method can converge in the linear-quadratical rate even with the stochastic approximation for the Hessian matrix and gradients.

1.3 Black-box optimization for hyper-parameter tuning

Finally, we consider a class of optimization problems, which often arise in hyper-parameter tuning for machine learning

$$\max_{\mathbf{x}\in\Omega} f(\mathbf{x}) \tag{1.5}$$

where Ω is a bounded hyper-rectangle, i.e., $\Omega = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ for some given $\mathbf{l}, \mathbf{u} \in \mathbb{R}^p$.

The Bayesian Optimization (`BO`) is a popular algorithm in the machine learning community and builds a surrogate for the objective and quantifies the uncertainty in that surrogate using a Bayesian machine learning technique, Gaussian process regression, and then uses an acquisition function defined from this surrogate to decide where to sample. On the other hand, the DIRECT algorithm in [53] is well-known for its simplicity and effectiveness in black-box optimization. However, it requires extensive function evaluations which makes it unsuitable in the machine learning application.

In Chapter 5, a new sampling efficient algorithm named Bayesian DIRECT (`BD`) is proposed by combining the benefits from the `BO` and the DIRECT algorithm. We conduct numerical experiments

on benchmark synthetic functions and machine learning algorithms including hyperparameter tuning for random forest, logistic regression and neural network. `BD` shows the faster convergence than the `BO` and DIRECT in most examples.

CHAPTER 2: Augmented Decomposition Algorithm

2.1 Introduction

Consider the following convex optimization problem of minimizing the sum of K separable, potentially nonsmooth convex functions subject to the linear constraints

$$\begin{aligned} \min_{x} \quad & f(x) = f_1(x_1) + \cdots + f_K(x_K) \\ \text{s.t.} \quad & Ex = E_1x_1 + \cdots + E_Kx_K = q, \\ & x_k \in X_k, \quad k = 1, 2, \ldots, K, \end{aligned} \tag{2.1}$$

where every f_k is a closed proper convex function (possibly nonsmooth) and each X_k is a closed convex set in $\mathbb{R}^{n_k}$. Let $x = (x_1, \ldots, x_K) \in \mathbb{R}^n$ be a partition of the variable x and $X = X_1 \times \cdots \times X_K \subset \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_K} = \mathbb{R}^n$ be the domain of x. For the linear constraint, $E = (E_1, \ldots, E_K) \in \mathbb{R}^{m \times n}$ is a partition of the matrix E consistent with the partition of x and $q \in \mathbb{R}^m$ is a column vector. A linear inequality constraint of the form $Ex \leq q$ can be easily transformed to the equality case by introducing a slack variable $x_{K+1} \geq 0$.

Optimization problems in the form of (2.1) arise in many application areas such as signal processing, statistics and machine learning. [65] summarizes a list of applications arising from many areas when more than two blocks are involved ($K \geq 3$).

Many decomposition algorithms have been proposed to solve the above optimization problem; see [15, 16, 18, 27, 69, 89, 101] and references therein. Among them, the ADMM method is perhaps the most popular approach to solve the decomposition problem due to its suitable parallel implementation and outstanding computational performance. When $K = 2$, the convergence of the ADMM was well studied in the framework of Douglas-Rachford splitting method [27]. The paper [24] proved the linear convergence of the ADMM when at least one of $f_i(\cdot)$ is strongly convex and E satisfies some additional assumptions. For the $K \geq 3$ case, it was shown in [37] that the global

convergence is guaranteed if all objective functions f_k are strongly convex. However, for general convex objective functions, it is acknowledged that the direct extension of the original ADMM may diverge [17]. Therefore, most recent researches have been focused on either analyzing problems with additional assumptions or showing the convergence results for variants of the ADMM; see [47, 103].

As an alternative to the ADMM algorithm for multi-block convex optimization problems, a new primal-dual algorithm called the *augmented decomposition algorithm* (ADA) was introduced in [82]. This method is closely related to the decomposition algorithm based on the partial inverses proposed in [89] but is derived from the proximal saddle point algorithm (PSPA) which is associated with a special primal-dual saddle function. It was shown in [82] that the algorithm is guaranteed to converge on the basis of convergence results of the proximal point algorithm (PPA) in [80]. What is more exciting is that the calculation of each iteration in PSPA can be carried out in parallel and its parallel implementation leads to the ADA.

Although the global convergence result for the ADA has been well studied under a general condition, the convergence rate result remained unknown. In the first part of this chapter, we focus on the convergence analysis of the ADA applied to problem (2.1). For that, we first provide a detailed proof for its convergence. Then, we show the $O(1/\nu)$ convergence rate in an ergodic sense. Finally, we improve the convergence result from $O(1/\nu)$ to $o(1/\nu)$ in a non-ergodic sense. These ideas are inspired by recent works on the ADMM and variants of the proximal method of multiplier [23, 43, 88].

Then, we consider the inexact ADA (iADA) in the second part. We first establish the global convergence result under certain approximation criteria. Then, under some mild assumptions on the function f_k and the structure of feasible set X_k, we show the local linear convergence of the iADA. This work is invoked by recent convergence rate results for the ADMM algorithm in [24, 47]. However, our proof is different from them in which we show the stability of a maximal monotone operator associated with the saddle function for a variant of (2.1). Denote the Lagrangian function by L for (2.1):

$$L(x,y) = \begin{cases} f(x) + \langle Ex - q, y\rangle, & \forall (x,y) \in X \times \mathbb{R}^m, \\ \infty, & \forall x \notin X. \end{cases} \tag{2.2}$$

The corresponding maximal monotone operator $\mathcal{T}_L$ [80] is defined by

$$\mathcal{T}_L(x,y) = \{(u,v)|(u,-v) \in \partial L(x,y)\} \tag{2.3}$$

where $\partial L(x,y)$ denotes the subgradient of the convex-concave function L. The inverse of $\mathcal{T}_L$ is given by

$$\mathcal{T}_L^{-1}(u,v) = \{(x,y)|(u,-v) \in \partial L(x,y)\}. \tag{2.4}$$

A solution to $(0,0) \in \mathcal{T}_L(x,y)$ is a saddle point of L. Classical convergence rate results for PPA [81] rely on the assumption that $\mathcal{T}_L^{-1}$ is Lipschitz continuous at $(0,0)$. This result was extended in [64] for situations in which $\mathcal{T}_L^{-1}(0,0)$ is not a singleton and the following holds:

$$\exists a > 0, \quad \exists \delta > 0 : \quad \forall w \in \mathcal{B}((0,0),\delta), \quad \forall z \in \mathcal{T}_L^{-1}w, \quad dist(z, \mathcal{T}_L^{-1}(0,0)) \leq a||w||. \tag{2.5}$$

It has been pointed out in many works that understanding the Lipschitzian behavior of $\mathcal{T}_L^{-1}$ at the origin is crucial to the study of the local convergence results for algorithms in the PPA framework; see [21, 35, 54, 58]. For instance, [54] showed the *metric subregularity* defined in [26] of $\mathcal{T}_L$ which is closely related to (2.5) under the so-called second order sufficient condition. However, this result inherently requires the solution uniqueness for problem (2.1). Compared with those assumptions, our assumptions in this part mainly rely on the polyhedral property of the feasible set X and the optimal solution set for (2.1) needs not to be a singleton. Our proof is based on Robinson's celebrated work on the error bound result for polyhedral multifunctions [78] and uses some ideas in the analysis for the satisfaction of a certain error bound condition in [47, 63].

Organization The remainder of this chapter is organized as follows. Section 2.2 first summarizes the basic idea of the proximal saddle point algorithm and its implementation, the ADA. Then, we show the convergence result for the ADA and compare it with the ADMM. In Section 2.3, we introduce the iADA and make some basic assumptions on the problem (2.1) for further discussion. Section 2.4 studies the stability results of the maximal monotone operator $\mathcal{T}_{\bar{L}}$. Section 2.5 establishes the global convergence and local linear convergence rate results of the iADA. Finally, some numerical examples are presented in Section 2.6 to demonstrate the performance the ADA and iADA.

Notation We use $\langle \cdot, \cdot \rangle$ and $||\cdot||$ to denote the standard inner product and $\mathcal{L}_2$-norm in the Euclidean space respectively. For any positive definite matrix $G \in S^n_{++}$ and $x, y \in \mathbb{R}^n$, the inner product $\langle x, y \rangle_G$ is defined by $x^T G y$ and its induced norm is denoted by $||\cdot||_G$. For $1 \leq q \leq \infty$, $||\cdot||_q$ represents the $\mathcal{L}_q$-norm. For any $E \in \mathbb{R}^{m \times n}$, $||E||$ denotes the spectral norm, *i.e.*, the largest singular value of E. For any function f, let $\mathrm{dom} f$ be the effective domain of the function f and $\mathrm{int}(\mathrm{dom} f)$ be the interior of $\mathrm{dom} f$. For any point $x \in \mathbb{R}^n$ and a closed convex set $C \subset \mathbb{R}^n$, $dist(x, C) = \min_{y \in C} ||y - x||$.

2.2 Global convergence of the ADA

In this chapter, we make the following standard assumption.

Assumption 2.1. *The global minimum of* (2.1) *is attainable and*

$$\mathrm{int}(X) \cap \mathrm{dom} f \cap \{x | Ex = q\} \neq \emptyset. \tag{2.6}$$

If X is polyhedral, an alternative assumption for (2.6) *can be that*

$$X \cap \mathrm{int}(\mathrm{dom} f) \cap \{x | Ex = q\} \neq \emptyset. \tag{2.7}$$

Assumption 2.1 guarantees the existence of a saddle point of L. Namely, there exist $\bar{x}$ and $\bar{y}$ such that

$$\bar{x} \in \underset{x \in X}{\mathrm{argmin}}\, L(x, \bar{y}), \qquad \bar{y} \in \underset{y \in \mathbb{R}^m}{\mathrm{argmax}}\, L(\bar{x}, y). \tag{2.8}$$

The dual function for problem (2.1) is

$$d(y) = \min_{x \in X} L(x, y) = \min_{x \in X} \{f(x) + \langle y, Ex - q \rangle\} \tag{2.9}$$

and its associated dual problem is given by

$$\max_{y \in \mathbb{R}^m} d(y). \tag{2.10}$$

Let X^* and Y^* be the optimal solution sets of (2.1) and (2.10) respectively. The set of saddle points for the Lagrangian (2.2) is given by $X^* \times Y^*$.

2.2.1 Augmented Decomposition Algorithm

Here, we first summarize the basic idea of PSPA and its parallel implementation, the ADA. For that, the original problem (2.1) is equivalently transformed into

$$\begin{aligned} \min_{x,w} \quad & f(x) = f_1(x_1) + \cdots + f_K(x_K) \\ \text{s.t.} \quad & E_j x_j - w_j = 0, \quad j = 1, \ldots, K-1, \\ & E_K x_K - q - w_K = 0, \\ & w_1 + \cdots + w_K = 0, \\ & x_k \in X_k, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{2.11}$$

If $x = (x_1, \ldots, x_K) \in \mathbb{R}^n$ is an optimal solution of (2.1), then

$$(x, w) = (x_1, \ldots, x_K, E_1 x_1, \ldots, E_{K-1} x_{K-1}, E_K x_K - q)$$

will be an optimal solution to (2.11). Instead of adding a multiplier vector for $w_1 + \cdots + w_K = 0$, [82] introduced W as a subspace of $(\mathbb{R}^m)^K$ which is defined as

$$W = \{w = (w_1, \ldots, w_K) | w_1 + \cdots + w_K = 0\} \subset (\mathbb{R}^m)^K. \tag{2.12}$$

The orthogonal complement subspace of W is given by

$$W^\perp = \{w = (w_1, \ldots, w_K) | w_1 = \cdots = w_K\} \subset (\mathbb{R}^m)^K. \tag{2.13}$$

For any $w = (w_1, \ldots, w_K) \in (\mathbb{R}^m)^K$, we use $P_{W^\perp}(w)$ to denote the projection of w onto the subspace $W^\perp$. In [82], the author proposed to add increments $u_i \in \mathbb{R}^m, i = 1, \ldots, K$ to the first K linear constraints in (2.11) and in addition, add to $w \in W$ a perturbation $v \in W^\perp$. The Lagrangian function associated with this perturbation finally works out in terms of the subspace

$$S = \{(\eta, \zeta) | P_{W^\perp}(\eta) = \zeta\} \subseteq (\mathbb{R}^m)^K \times W^\perp, \tag{2.14}$$

and the functions

$$L_j(x_j, \eta_j) = \begin{cases} f_j(x_j) + \eta_j \cdot E_j x_j, \text{ if } j = 1, \ldots, K-1, \\ f_K(x_K) + \eta_K \cdot (E_K x_K - q), \text{ o.w.} \end{cases} \tag{2.15}$$

to mean that

$$\bar{L}(w, x, \eta, \zeta) = \begin{cases} \sum_{j=1}^{K} [L_j(x_j, \eta_j) - \eta_j \cdot w_j], \text{ if } (w, x) \in W \times X, (\eta, \zeta) \in S, \\ -\infty, \text{ if } (w, x) \in W \times X, (\eta, \zeta) \notin S, \\ +\infty, \text{ if } (w, x) \notin W \times X. \end{cases} \tag{2.16}$$

The next lemma shows the relationship between $L(x, y)$ and $\bar{L}(w, x, \eta, \zeta)$.

Lemma 2.1. *If $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ is a saddle point of the Lagrangian function in* (2.16)*, then $\bar{\eta}_1 = \bar{\eta}_2 = \cdot = \bar{\eta}_K$ and $(\bar{x}, \bar{\eta}_1)$ is a saddle point of* (2.2)*. Conversely, let $(\bar{x}, \bar{y})$ be a saddle point of* (2.2)*, and define $\bar{w} = (E_1\bar{x}_1, \ldots, E_{K-1}\bar{x}_{K-1}, E_K\bar{x}_K - q) \in (\mathbb{R}^m)^K$, $\bar{\eta} = (\bar{y}, \ldots, \bar{y}) \in (\mathbb{R}^m)^K$ and $\bar{\zeta} = \bar{\eta}$. Then $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ is a saddle point of* (2.16)*.*

Proof. The dual problem associated with (2.16) is

$$\max_{(\eta,\zeta)\in S} \{\bar{g}(\eta, \zeta) = \inf_{(w,x)\in W\times X} \bar{L}(w, x, \eta, \zeta)\} \tag{2.17}$$

with its feasible set given by

$$\{(\eta, \zeta) | \bar{g}(\eta, \zeta) > -\infty\} \subset S.$$

As $w \cdot \eta$ cannot be ∞, this implies $\eta_1 = \eta_2 = \cdot = \eta_K$. As a consequence, the dual problem reduces to

$$\max_{(\eta,\zeta)\in S} \{\bar{g}(\eta, \zeta) = \inf_{x\in X} f(x) + \langle \eta_1, Ex - q \rangle\} \tag{2.18}$$

which is equivalent to the dual problem corresponding to (2.2). So we can conclude the first part. The second part is similarly based on the above observation for the dual whose proof is omitted here. □

Based on [80], the proximal method of multipliers is derived by adding both primal and dual proximal terms into the Lagrangian (2.16). More explicitly, the proximal saddle point algorithm in [82] can be described as the following:

Generate a sequence of elements $(w^\nu, x^\nu) \in W \times X$ and $(\eta^\nu, \zeta^\nu) \in S$ by letting

$$\bar{L}^\nu(w,x,\eta,\zeta) = \bar{L}(w,x,\eta,\zeta) + \frac{\rho}{2}||w-w^\nu||^2 + \frac{1}{2c}||x-x^\nu||^2 - \frac{1}{2\rho}||\eta-\eta^\nu||^2 - \frac{1}{2\rho}||\zeta-\zeta^\nu||^2 \tag{2.19}$$

and calculating

$$(w^{\nu+1}, x^{\nu+1}, \eta^{\nu+1}, \zeta^{\nu+1}) = \text{ unique saddle point of } \bar{L}^\nu(w,x,\eta,\zeta)$$

with respect to minimizing over $(w,x) \in W \times X$ and maximizing over $(\eta,\zeta) \in S$. According to [80], the sequence $(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ generated by the above algorithm from any initial $(w^1, x^1) \in W \times X$ and $(\eta^1, \zeta^1) \in S$ is certain to converge to some saddle point $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ of the Lagrangian $\bar{L}$. With the special structure of the saddle point problem, the calculation of the saddle point in (2.19) can be carried out in the following parallel algorithm ADA. For simplicity, we denote

$$\phi^\nu_{k,\rho,c}(x_k) = \begin{cases} f_k(x_k) + \frac{\rho}{4}||E_k x_k - w_k^\nu + \frac{2}{\rho}y_k^\nu||_2^2 + \frac{1}{2c}||x_k - x_k^\nu||_2^2, & k = 1,\dots,K-1, \\ f_K(x_K) + \frac{\rho}{4}||E_K x_K - q - w_K^\nu + \frac{2}{\rho}y_K^\nu||_2^2 + \frac{1}{2c}||x_K - x_K^\nu||_2^2, & k = K. \end{cases} \tag{2.20}$$

Algorithm 2.1 Augmented decomposition algorithm

1: Given $w^0 \in W, x^0 \in X, y^0 \in (\mathbb{R}^m)^K$
2: **for** $\nu = 0, 1, \dots$ **do**
3: $\quad x_k^{\nu+1} = \text{argmin}_{x_k \in X_k}\, \phi^\nu_{k,\rho,c}(x_k), k = 1,\dots,K$
4: $\quad \eta_k^{\nu+1} = \begin{cases} y_k^\nu + \frac{\rho}{2}[E_k x_k^{\nu+1} - w_k^\nu], \text{ if } k = 1,\dots,K-1 \\ y_K^\nu + \frac{\rho}{2}[E_K x_K^{\nu+1} - q - w_K^\nu], \text{ if } k = K \end{cases}$
5: $\quad$ **for** $k = 1,\dots,K$ **do**
6: $\quad\quad \zeta_k^{\nu+1} = \frac{1}{K}\sum_{j=1}^K \eta_j^{\nu+1}$
7: $\quad\quad w_k^{\nu+1} = w_k^\nu + \frac{1}{\rho}[\eta_k^{\nu+1} - \zeta_k^{\nu+1}]$
8: $\quad\quad y_k^{\nu+1} = \frac{1}{2}[\eta_k^{\nu+1} + \zeta_k^{\nu+1}]$

2.2.2 Convergence of the ADA

In this subsection, we assume $q = 0$ for notational simplicity which will not influence the proofs below. Define the matrix

$$G := \begin{pmatrix} \rho I_{mK} & & & \\ & \frac{1}{c} I_n & & \\ & & \frac{1}{\rho} I_{mK} & \\ & & & \frac{1}{\rho} I_{mK} \end{pmatrix}. \tag{2.21}$$

Hence $G \succ 0$ and $||\cdot||_G$ defines a norm. Let $\hat{u} = (\hat{w}, \hat{x}, \hat{\eta}, \hat{\zeta})$ and $u^\nu = (w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ where $\hat{u}$ is a saddle point of the Lagrangian function (2.16) and u^ν is the current iteration point. The convergence result for ADA was established in [82] on the basis of convergence results for the classic PPA. Here, we import the result and provide an alternative proof for it.

Theorem 2.1. *Under Assumption 2.1, for any $\rho > 0$ and $c > 0$, the sequence $\{(w^\nu, x^\nu, y^\nu)\}_{\nu=1}^{\infty}$ generated in $W \times X \times (\mathbb{R}^m)^K$ by the ADA from any starting point converges to some $(\bar{w}, \bar{x}, \bar{y})$ such that*

(a) $(\bar{w}, \bar{x})$ solves (2.11)*, hence $\bar{x}$ solves* (2.1)*,*

(b) $\bar{y}_1 = \cdots = \bar{y}_q \in \mathbb{R}^m$, and this common multiplier vector solves (2.10)*.*

Proof. From Assumption 2.1 and Lemma 2.1, there exists a saddle point $(\hat{w}, \hat{x}, \hat{\eta}, \hat{\zeta}) \in W \times X \times S$ of the Lagrangian function (2.16). For each iteration $\nu + 1$, due to the minimax operation on (2.19), from the primal perspective, we have the following inequality

$$\begin{aligned} &\sum_{k=1}^{K} f_k(x_k) + \sum_{k=1}^{K} \langle \eta_k^{\nu+1}, E_k x_k - w_k \rangle \\ \geq &\sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \eta_k^{\nu+1}, E_k x_k^{\nu+1} - w_k^{\nu+1} \rangle + \frac{1}{c} \sum_{k=1}^{K} \langle x_k - x_k^{\nu+1}, x_k^{\nu} - x_k^{\nu+1} \rangle \\ &+ \rho \sum_{k=1}^{K} \langle w_k - w_k^{\nu+1}, w_k^{\nu} - w_k^{\nu+1} \rangle \end{aligned} \tag{2.22}$$

for any $x \in X$ and $w \in W$. Applying $(w, x) = (\hat{w}, \hat{x})$ to (2.22) and noticing that $E\hat{x}_k = \hat{w}_k, k =$

$1, \ldots, K$, we obtain

$$\begin{aligned}\min P := \sum_{k=1}^{K} f_k(\hat{x}_k) \geq \sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \eta_k^{\nu+1}, E_k x_k^{\nu+1} - w_k^{\nu+1} \rangle \\ - \frac{1}{c} \sum_{k=1}^{K} \langle x_k^{\nu+1} - \hat{x}_k, x_k^{\nu} - x_k^{\nu+1} \rangle - \rho \sum_{k=1}^{K} \langle w_k^{\nu+1} - \hat{w}_k, w_k^{\nu} - w_k^{\nu+1} \rangle.\end{aligned} \tag{2.23}$$

Similarly, from the dual perspective and the saddle-point property of $(\hat{w}, \hat{x}, \hat{\eta}, \hat{\zeta})$, the following inequality

$$\begin{aligned}\min P = &\sum_{k=1}^{K} f_k(\hat{x}_k) \leq \sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \hat{\eta}_k, E_k x_k^{\nu+1} - w_k^{\nu+1} \rangle \\ \leq &\sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \eta_k^{\nu+1}, E_k x_k^{\nu+1} - w_k^{\nu+1} \rangle \\ &+ \frac{1}{\rho} \sum_{k=1}^{K} \langle \eta_k^{\nu+1} - \hat{\eta}_k, \eta_k^{\nu} - \eta_k^{\nu+1} \rangle + \frac{1}{\rho} \sum_{k=1}^{K} \langle \zeta_k^{\nu+1} - \hat{\zeta}_k, \zeta_k^{\nu} - \zeta_k^{\nu+1} \rangle\end{aligned} \tag{2.24}$$

holds. Combining the above two inequalities with the following identity

$$2\langle a - b, c - a \rangle = ||c - b||_2^2 - ||c - a||_2^2 - ||b - a||_2^2, \tag{2.25}$$

we have

$$\begin{aligned}&\sum_{k=1}^{K} \left(\frac{1}{c} ||x_k^{\nu} - \hat{x}_k||_2^2 + \rho ||w_k^{\nu} - \hat{w}_k||_2^2 + \frac{1}{\rho} ||\eta_k^{\nu} - \hat{\eta}_k||_2^2 + \frac{1}{\rho} ||\zeta_k^{\nu} - \hat{\zeta}_k||_2^2\right) \\ - &\sum_{k=1}^{K} \left(\frac{1}{c} ||x_k^{\nu+1} - \hat{x}_k||_2^2 + \rho ||w_k^{\nu+1} - \hat{w}_k||_2^2 + \frac{1}{\rho} ||\eta_k^{\nu+1} - \hat{\eta}_k||_2 + \frac{1}{\rho} ||\zeta_k^{\nu+1} - \hat{\zeta}_k||_2^2\right) \\ \geq &\sum_{k=1}^{K} \left(\frac{1}{c} ||x_k^{\nu+1} - x_k^{\nu}||_2^2 + \rho ||w_k^{\nu+1} - w_k^{\nu}||_2^2 + \frac{1}{\rho} ||\eta_k^{\nu+1} - \eta_k^{\nu}||_2^2 + \frac{1}{\rho} ||\zeta_k^{\nu+1} - \zeta_k^{\nu}||_2^2\right)\end{aligned} \tag{2.26}$$

which is equivalent with

$$||u^{\nu} - \hat{u}||_G^2 - ||u^{\nu+1} - \hat{u}||_G^2 \geq ||u^{\nu} - u^{\nu+1}||_G^2. \tag{2.27}$$

From this inequality, we can easily conclude that

(i) $\sum_{\nu=0}^{\infty} ||u^\nu - u^{\nu+1}||_G^2 < \infty$;

(ii) $\{u^\nu = (w^\nu, x^\nu, \eta^\nu, \zeta^\nu)\}$ lies in a compact region;

(iii) $||u^\nu - \hat{u}||_G$ is a monotonically non-increasing sequence and thus converges.

From (ii), by passing to a subsequence if necessary, there exists at least one limiting point of $\{(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)\}$, denoted as $\bar{u} = (\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$. It follows from (i) that $x^\nu - x^{\nu+1} \to 0$, $w^\nu - w^{\nu+1} \to 0$ and $\eta^\nu - \eta^{\nu+1} \to 0$. The update rule for w implies that $\bar{\eta}_1 = \cdots = \bar{\eta}_K$ and thus $\bar{y}_1 = \cdots = \bar{y}_K = \bar{\eta}_1$. Since $\eta_k^{\nu+1} = y_k^\nu + \frac{\rho}{2}[E_k x_k^{\nu+1} - w_k^\nu]$, $E_k \bar{x}_k = \bar{w}_k$ holds and thus $E\bar{x} = 0$ which implies the feasibility of $\bar{x}$. Due to the optimality condition for each block in iteration $\nu + 1$, we have

$$0 \in \partial f_k(x_k^{\nu+1}) + E_k^T \eta_k^{\nu+1} + \frac{1}{c}(x_k^{\nu+1} - x_k^\nu) + N_{X_k}(x_k^{\nu+1}), \quad k = 1, \ldots, K.$$

By passing to the limit, we obtain

$$0 \in \partial f(\bar{x}) + E^T \bar{\eta}_1 + N_X(\bar{x}).$$

As a result, $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ is a saddle point of the Lagrangian function (2.16). Next, we show the uniqueness of the limit point to complete the proof. Let $\bar{u}^1 = (\bar{w}^1, \bar{x}^1, \bar{\eta}^1, \bar{\zeta}^1)$ and $\bar{u}^2 = (\bar{w}^2, \bar{x}^2, \bar{\eta}^2, \bar{\zeta}^2)$ be any two different limit points of $u^\nu = (w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$. By the previous argument, both of them are saddle points of (2.16). From (iii), we know the existence of the following limits

$$\lim_{\nu \to \infty} ||u^\nu - \bar{u}^i||_G = \beta_i, \quad i = 1, 2.$$

With the following equality

$$||u^\nu - \bar{u}^1||_G^2 - ||u^\nu - \bar{u}^2||_G^2 = -2\langle u^\nu, \bar{u}^1 - \bar{u}^2 \rangle_G + ||\bar{u}^1||_G^2 - ||\bar{u}^2||_G^2$$

and by passing to the limit, we have

$$\beta_1^2 - \beta_2^2 = -2\langle \bar{u}^1, \bar{u}^1 - \bar{u}^2 \rangle_G + ||\bar{u}^1||_G^2 - ||\bar{u}^2||_G^2 = -||\bar{u}^1 - \bar{u}^2||_G^2$$

and

$$\beta_1^2 - \beta_2^2 = -2\langle \bar{u}^2, \bar{u}^1 - \bar{u}^2 \rangle_G + ||\bar{u}^1||_G^2 - ||\bar{u}^2||_G^2 = ||\bar{u}^1 - \bar{u}^2||_G^2.$$

Thus we obtain $||\bar{u}^1 - \bar{u}^2||_G = 0$ which implies that the sequence $(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ converges to some saddle point of the Lagrangian function (2.16) and hence (a) and (b) hold. □

2.2.3 Rate of Convergence

In this subsection, we study the global convergence rate for the ADA. We first show the sublinear convergence result of the ADA in an ergodic sense. The proof follows the same idea as that in [88].

Theorem 2.2. *Let $\{u^\nu = (w^\nu, x^\nu, \eta^\nu, \zeta^\nu)\}$ in $W \times X \times S$ be the infinite sequence generated by the ADA. For any integer $N > 0$, define $\tilde{x}_N$ by*

$$\tilde{x}_N = \frac{1}{N} \sum_{\nu=1}^{N} x^\nu.$$

Then for any saddle point $\hat{u} = (\hat{w}, \hat{x}, \hat{\eta}, \hat{\zeta}) \in W \times X \times S$ of (2.16),

$$f(\tilde{x}_N) + \langle \hat{\eta}_1, E\tilde{x}_N \rangle - \min P \leq \frac{||\hat{u} - u^0||_G^2}{N}.$$

Proof. For any saddle point $(\hat{w}, \hat{x}, \hat{\eta}, \hat{\zeta}) \in W \times X \times S$ of the Lagrangian function (2.16), it follows from (2.23) and (2.24) that

$$\begin{aligned} & ||u^\nu - \hat{u}||_G^2 - ||u^{\nu+1} - \hat{u}||_G^2 \\ \geq & ||u^\nu - u^{\nu+1}||_G^2 + \sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \hat{\eta}_k, E_k x_k^{\nu+1} \rangle - \min P \\ \geq & \sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \hat{\eta}_k, E_k x_k^{\nu+1} \rangle - \min P. \end{aligned} \tag{2.28}$$

Summing (2.28) for $\nu = 0, 1, \ldots, N-1$, we obtain

$$\begin{aligned} & ||u^0 - \hat{u}||_G^2 \\ \geq & \sum_{\nu=0}^{N-1} \{ \sum_{k=1}^{K} f_k(x_k^{\nu+1}) + \sum_{k=1}^{K} \langle \hat{\eta}_k, E_k x_k^{\nu+1} \rangle \} - N \min P \\ \geq & N[f(\tilde{x}_N) + \langle \hat{\eta}_1, E\tilde{x}_N \rangle - \min P] \end{aligned} \tag{2.29}$$

where the second inequality results from the convexity of $f(\cdot)$ and the fact $\hat{\eta}_1 = \hat{\eta}_2 = \cdots = \hat{\eta}_K$. The assertion (2.2) follows immediately from the above inequality. □

Next, we shall prove the $o(1/\nu)$ convergence of the ADA. Motivated by [23, 44], we use the quantity $||u^\nu - u^{\nu+1}||_G^2$ as a measure of the convergence rate. In fact, if $||u^\nu - u^{\nu+1}||_G^2 = 0$, then $u^{\nu+1}$ is an optimal solution, *i.e.*, $(x^{\nu+1}, \eta_1^{\nu+1}) \in X^* \times Y^*$. More explicitly, $||u^\nu - u^{\nu+1}||_G^2 = 0$ implies the following:

$$x^\nu = x^{\nu+1} \text{ and } w^\nu = w^{\nu+1}. \tag{2.30}$$

By the update step for w, we can conclude $\eta_1^{\nu+1} = \cdots = \eta_K^{\nu+1}$. Combining this with $x^\nu = x^{\nu+1}$, we obtain

$$0 \in \partial f(x^{\nu+1}) + E^T \eta_1^{\nu+1} + N_X(x^{\nu+1}), \tag{2.31}$$

or equivalently, $(x^{\nu+1}, \eta_1^{\nu+1}) \in X^* \times Y^*$. Conversely, if the quantity $||u^\nu - u^{\nu+1}||_G^2$ is relatively large, $u^{\nu+1}$ should not be close to the optimal solution set. Based on previous analysis, $||u^\nu - u^{\nu+1}||_G^2$ is a reasonable measure to quantify the distance between $u^{\nu+1}$ and the optimal solution set.

To show the convergence rate, we first prove the following lemma on the monotonicity property of the iterations:

Lemma 2.2. *Let u^ν be defined as in Theorem 2.2. Then*

$$||u^\nu - u^{\nu+1}||_G^2 \leq ||u^{\nu-1} - u^\nu||_G^2.$$

Proof. For notational simplicity, for each iteration ν, we introduce

$$\Delta u^{\nu+1} = \begin{pmatrix} \Delta w^{\nu+1} \\ \Delta x^{\nu+1} \\ \Delta \eta^{\nu+1} \\ \Delta \zeta^{\nu+1} \end{pmatrix} = \begin{pmatrix} w^{\nu} - w^{\nu+1} \\ x^{\nu} - x^{\nu+1} \\ \eta^{\nu} - \eta^{\nu+1} \\ \zeta^{\nu} - \zeta^{\nu+1} \end{pmatrix}. \tag{2.32}$$

By the optimality of $x_k^{\nu+1}$ in iteration $\nu+1$ and the update rule of $\eta_k^{\nu+1}$, we have

$$\frac{1}{c}(x_k^{\nu} - x_k^{\nu+1}) - E_k^T \eta_k^{\nu+1} \in \partial f_k(x_k^{\nu+1}) + N_{X_k}(x_k^{\nu+1}), \qquad k = 1, \ldots, K. \tag{2.33}$$

Considering the ν-th and $\nu+1$-th iteration, such optimality yields

$$\underbrace{\frac{1}{c}\langle \Delta x_k^{\nu+1}, \Delta x_k^{\nu} - \Delta x_k^{\nu+1} \rangle}_{(a)} - \langle E_k \Delta x_k^{\nu+1}, \Delta \eta_k^{\nu+1} \rangle \geq 0, \qquad k = 1, \ldots, K. \tag{2.34}$$

For the second term in the above inequality,

$$
\begin{aligned}
&-\sum_{k=1}^{K}\langle E_k\Delta x_k^{\nu+1},\Delta\eta_k^{\nu+1}\rangle=\sum_{k=1}^{K}\langle E_k x_k^{\nu+1}-E_k x_k^{\nu},\Delta\eta_k^{\nu+1}\rangle\\
=&\sum_{k=1}^{K}\langle\frac{\eta_k^{\nu+1}-y_k^{\nu}}{\rho/2}-\frac{\eta_k^{\nu}-y_k^{\nu-1}}{\rho/2}+w_k^{\nu}-w_k^{\nu-1},\Delta\eta_k^{\nu+1}\rangle\\
=&\sum_{k=1}^{K}\langle\frac{\eta_k^{\nu+1}-\frac{\eta_k^{\nu}+\zeta_k^{\nu}}{2}}{\rho/2}-\frac{\eta_k^{\nu}-\frac{\eta_k^{\nu-1}+\zeta_k^{\nu-1}}{2}}{\rho/2}+\frac{\eta_k^{\nu}-\zeta_k^{\nu}}{\rho},\Delta\eta_k^{\nu+1}\rangle\\
=&\sum_{k=1}^{K}\frac{1}{\rho}\langle\Delta\eta_k^{\nu}-\Delta\eta_k^{\nu+1},\Delta\eta_k^{\nu+1}\rangle+\sum_{k=1}^{K}\frac{1}{\rho}\langle\Delta\zeta_k^{\nu},\Delta\eta_k^{\nu+1}\rangle+\\
&\sum_{k=1}^{K}\frac{1}{\rho}\langle\eta_k^{\nu+1}-\eta_k^{\nu},\eta_k^{\nu}-\eta_k^{\nu+1}\rangle+\sum_{k=1}^{K}\frac{1}{\rho}\langle\eta_k^{\nu}-\zeta_k^{\nu},\Delta\eta_k^{\nu+1}\rangle\\
=&\underbrace{\sum_{k=1}^{K}\frac{1}{\rho}\langle\Delta\eta_k^{\nu}-\Delta\eta_k^{\nu+1},\Delta\eta_k^{\nu+1}\rangle}_{(b)}+\underbrace{\sum_{k=1}^{K}\frac{1}{\rho}\langle\Delta\zeta_k^{\nu}-\Delta\zeta_k^{\nu+1},\Delta\zeta_k^{\nu+1}\rangle}_{(c)}+\\
&\underbrace{\sum_{k=1}^{K}\frac{1}{\rho}\langle\eta_k^{\nu+1}-\zeta_k^{\nu+1},\eta_k^{\nu}-\eta_k^{\nu+1}\rangle}_{(d)}.
\end{aligned}
\tag{2.35}
$$

For term (d),

$$
\begin{aligned}
&2\sum_{k=1}^{K}\frac{1}{\rho}\langle\eta_k^{\nu+1}-\zeta_k^{\nu+1},\eta_k^{\nu}-\eta_k^{\nu+1}\rangle\\
=&\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu}-\zeta_k^{\nu+1}||_2^2-\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu}-\eta_k^{\nu+1}||_2^2-\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu+1}-\zeta_k^{\nu+1}||_2^2\\
=&\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu}-\zeta_k^{\nu}+\zeta_k^{\nu}-\zeta_k^{\nu+1}||_2^2-\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu}-\eta_k^{\nu+1}||_2^2-\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu+1}-\zeta_k^{\nu+1}||_2^2\\
=&\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu}-\zeta_k^{\nu}||_2^2+\underbrace{\sum_{k=1}^{K}\frac{2}{\rho}\langle\eta_k^{\nu}-\zeta_k^{\nu},\zeta_k^{\nu}-\zeta_k^{\nu+1}\rangle}_{=0}+\frac{1}{\rho}||\zeta^{\nu}-\zeta^{\nu+1}||_2^2-\\
&\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu}-\eta_k^{\nu+1}||_2^2-\sum_{k=1}^{K}\frac{1}{\rho}||\eta_k^{\nu+1}-\zeta_k^{\nu+1}||_2^2.
\end{aligned}
\tag{2.36}
$$

Applying the equality (2.25) to (a), (b) and (c) and combining them with the above transformation

for term (d), the inequality (2.34) yields

$$||\Delta u^{\nu}||_G^2 - ||\Delta u^{\nu+1}||_G^2 \geq \sum_{k=1}^{K} \frac{1}{c}||\Delta x_k^{\nu} - \Delta x_k^{\nu+1}||_2^2 + \sum_{k=1}^{K} \frac{1}{\rho}||\Delta \eta_k^{\nu} - \Delta \eta_k^{\nu+1}||_2^2 + \sum_{k=1}^{K} \frac{1}{\rho}||\Delta \zeta^{\nu} - \Delta \zeta^{\nu+1}||_2^2 + \underbrace{\sum_{k=1}^{K} \frac{1}{\rho}(||\eta_k^{\nu} - \eta_k^{\nu+1}||_2^2 - ||\zeta_k^{\nu} - \zeta_k^{\nu+1}||_2^2)}_{\geq 0} \geq 0. \tag{2.37}$$

The nonnegativity of the last term is a direct result of the definition $\zeta_1^{\nu} = \cdots = \zeta_K^{\nu} = \frac{1}{K}\sum_{j=1}^{K} \eta_j^{\nu}$ and CauchySchwarz inequality. Hence the inequality (2.2) holds. □

The following elementary lemma helps to improve the convergence rate from $O(1/\nu)$ to $o(1/\nu)$.

Lemma 2.3. *Suppose a sequence $\{a_\nu\}_{\nu=0}^{\infty} \subseteq \mathbb{R}$ satisfies the following: (a) $a_\nu \geq 0$; (b) $\sum_{\nu=0}^{\infty} a_\nu < \infty$; and (c) a_ν is monotonically non-increasing. Then, we have $a_\nu = o(1/\nu)$.*

Proof. See Lemma 1.1 in [23]. □

Combining the results from previous two lemmas, we present the $o(1/\nu)$ convergence of the ADA.

Theorem 2.3. *Let $\{u^{\nu} = (w^{\nu}, x^{\nu}, \eta^{\nu}, \zeta^{\nu})\}$ in $W \times X \times S$ be the infinite sequence generated by the ADA, then*

$$||u^{\nu} - u^{\nu+1}||_G^2 = o(1/\nu) \tag{2.38}$$

holds and thus

$$||x^{\nu} - x^{\nu+1}||_2^2 = o(1/\nu) \tag{2.39}$$

and

$$||\sum_{k=1}^{K} E_k x_k^{\nu+1}||_2^2 = o(1/\nu). \tag{2.40}$$

Proof. In the proof of Theorem 2.1, we have shown that

$$\sum_{\nu=0}^{\infty} ||u^{\nu} - u^{\nu+1}||_G^2 < \infty.$$

On the other hand, Lemma 2.2 proved the non-increasing property of $||u^{\nu} - u^{\nu+1}||_G^2$. Hence, (2.38) follows directly from Lemma 2.3 and then (2.39) holds. For the estimate for the constraint in (2.40),

we have

$$
\begin{aligned}
&||\sum_{k=1}^{K} E_k x_k^{\nu+1}||_2^2 = ||\sum_{k=1}^{K}(E_k x_k^{\nu+1} - w_k^{\nu})||_2^2 = \frac{4}{\rho^2}||\sum_{k=1}^{K}(\eta_k^{\nu+1} - y_k^{\nu})||_2^2 \\
\leq &\frac{4K}{\rho^2}\sum_{k=1}^{K}||\eta_k^{\nu+1} - \frac{1}{2}(\eta_k^{\nu} + \zeta_k^{\nu})||_2^2 \\
= &\frac{K}{\rho^2}\sum_{k=1}^{K}||\eta_k^{\nu+1} - \eta_k^{\nu} + \eta_k^{\nu+1} - \zeta_k^{\nu+1} + \zeta_k^{\nu+1} - \zeta_k^{\nu}||_2^2 \\
\leq &\frac{3K}{\rho^2}||\eta^{\nu+1} - \eta^{\nu}||_2^2 + 3K||w^{\nu+1} - w^{\nu}||_2^2 + \frac{3K}{\rho^2}||\zeta^{\nu+1} - \zeta^{\nu}||_2^2 = o(1/\nu),
\end{aligned}
\tag{2.41}
$$

where the first two equalities result from $w^{\nu} \in W$ and the updating rule for $\eta^{\nu+1}$. This finishes the proof for (2.40). □

From Theorem 2.3, a reasonable stopping criterion for the ADA can be either

$$
\frac{||x^{\nu} - x^{\nu+1}||}{\max\{1, ||x^{\nu}||\}} \leq \epsilon \tag{2.42}
$$

or

$$
\frac{||Ex^{\nu+1} - q||}{\max\{1, ||q||\}} \leq \epsilon \tag{2.43}
$$

for some given tolerance ϵ.

2.2.4 Relation to the ADMM

The ADA is closely related to the ADMM. Here, we compare the ADA with two variants of ADMM, namely, the Variable Splitting ADMM and the Proximal Jacobian ADMM. For simplicity of notation, we assume $q = 0$.

Applying the classical two-block ADMM to the transformation in (2.11), [103] proposed the following Variable Splitting ADMM (VSADMM), see Algorithm 2.2. The convergence result for

Algorithm 2.2 Variable Splitting ADMM

1: Given $w^0 \in W, x^0 \in X, y^0 \in (\mathbb{R}^m)^K, \beta > 0$
2: **for** $\nu = 0, 1, \ldots$ **do**
3: $\quad x_k^{\nu+1} = \text{argmin}_{x_k \in X_k} f_k(x_k) + \frac{\beta}{2}||E_k x_k - w_k^{\nu} + \frac{y_k^{\nu}}{\beta}||_2^2, \quad k = 1, \ldots, K,$
4: $\quad w^{\nu+1} = \text{argmin}_{w \in W} \frac{\beta}{2}\sum_{k=1}^{K}||E_k x_k^{\nu+1} - w_k + \frac{y_k^{\nu}}{\beta}||_2^2,$
5: $\quad y_k^{\nu+1} = y_k^{\nu} + \beta[E_k x_k^{\nu+1} - w_k^{\nu+1}], \quad k = 1, \ldots, K.$

VSADMM was established on the basis of the classical two-block ADMM. Compared to the ADA, we notice that no proximal terms exist during the x-update in the VSADMM. Therefore, the full column rank assumption of E_k is necessary for the VSADMM to guarantee the solution uniqueness in each iteration. The w-update step in the VSADMM also differs from that in the ADA as it does not use the information on the previous iteration explicitly.

The Proximal Jacobian ADMM (Prox-JADMM) provided in [23] solves problem (2.1) directly by adding a proximal term in the Jacobian-type ADMM, see Algorithm 2.3. It is worth noting

Algorithm 2.3 Proximal Jacobian ADMM

1: Given $x^0 \in X, \lambda^0 \in \mathbb{R}^m, \beta > 0$
2: **for** $\nu = 0, 1, \ldots$ **do**
3: **for** $k = 1, \ldots, K$ **do**
4: $x_k^{\nu+1} = \text{argmin}_{x_k \in X_k} f_k(x_k) + \frac{\beta}{2}||E_k x_k + \sum_{j \neq k} E_j x_j^{\nu} - \frac{\lambda^{\nu}}{\beta}||_2^2 + \frac{1}{2}||x_k - x_k^{\nu}||_{P_k}^2$,
5: $\lambda^{\nu+1} = \lambda^{\nu} - \gamma\beta \sum_{k=1}^{K} E_k x_k^{\nu+1}$,

that the ADA shares the same $o(1/\nu)$ convergence rate as the Prox-JADMM. However, the Prox-JADMM requires the constraints E_k, the proximal terms P_k and the damping parameter γ to satisfy certain relationships to guarantee the convergence. Because the convergence results for the ADA are established using a very different approach, we impose no restriction on the proximal terms.

2.3 The Inexact Augmented Decomposition Algorithm

Here, we first review the general convergence theory of the (inexact-)proximal point algorithm (PPA) developed in [80, 81]. Let $\mathcal{T} : \mathcal{X} \rightrightarrows \mathcal{X}$ be a maximally monotone operator. In order to solve the inclusion problem:

$$0 \in \mathcal{T}(z), \tag{2.44}$$

PPA takes the form of

$$z^{k+1} \approx (I + c_k \mathcal{T})^{-1}(z^k), \quad \forall k \geq 0, \tag{2.45}$$

in the $(k+1)$-th iteration with a given sequence $c_k \uparrow c_\infty \leq \infty$. The convergence result of PPA can be guaranteed as long as the approximation computation satisfies certain criteria; see [80, 81]. In addition, the local linear convergence result could be established when $\mathcal{T}^{-1}$ is Lipschitz continuous at the origin. In accordance with the PPA, the inexact version of the ADA comes out naturally as

follows in Algorithm 2.4. The iADA allows the subproblems to be solved inexactly which is very important in many applications as it might be very expensive to solve these subproblems exactly.

Algorithm 2.4 Inexact augmented decomposition algorithm

Given $w^0 \in W, x^0 \in X, y^0 \in (\mathbb{R}^m)^K$
for $\nu = 0, 1, \ldots$ **do**
  $x_k^{\nu+1} \approx \operatorname{argmin}_{x_k \in X_k} \phi_{k,\rho,c}^{\nu}(x_k), k = 1, \ldots, K$
  $\eta_k^{\nu+1} = \begin{cases} y_k^{\nu} + \frac{\rho}{2}[E_k x_k^{\nu+1} - w_k^{\nu}], \text{ if } k = 1, \ldots, K-1 \\ y_K^{\nu} + \frac{\rho}{2}[E_K x_K^{\nu+1} - q - w_K^{\nu}], \text{ if } k = K \end{cases}$
  for $k = 1, \ldots, K$ **do**
    $\zeta_k^{\nu+1} = \frac{1}{K} \sum_{j=1}^{K} \eta_j^{\nu+1}$
    $w_k^{\nu+1} = w_k^{\nu} + \frac{1}{\rho}[\eta_k^{\nu+1} - \zeta_k^{\nu+1}]$
    $y_k^{\nu+1} = \frac{1}{2}[\eta_k^{\nu+1} + \zeta_k^{\nu+1}]$

Two natural concerns arise for the iADA: (1) the global convergence and (2) the local convergence rate. For that, we make the following assumptions on f for the rest of the paper:

Assumption 2.2. *(a) $f = f_1(x_1) + \cdots + f_K(x_K)$, with each f_k given by*

$$f_k(x_k) = g_k(A_k x_k) + h_k(x_k) \tag{2.46}$$

where g_k and h_k are both closed proper convex functions and A_k's are some given matrices.

(b) Every g_k is strongly convex and continuously differentiable on $\operatorname{int}(\operatorname{dom} g_k)$ *with a Lipschitz continuous gradient*

$$||A_k^T \nabla g_k(A_k x_k) - A_k^T \nabla g_k(A_k x_k')|| \leq L_g^k ||A_k(x_k - x_k')||, \qquad \forall x_k, x_k' \in X_k \tag{2.47}$$

where $L_g^k \geq 0, k = 1, \ldots, K$.

(c) The epigraph of each h_k is a polyhedral convex set.

(d) The feasible sets $X_k, k = 1, \ldots, K$ are polyhedral convex sets.

(e) The feasible sets $X_k, k = 1, \ldots, K$ are compact sets.

Here are several comments on the above assumptions.

- Either g_k or h_k can be absent in f_k. Although g_k is assumed to be strongly convex, we do not impose any condition on A_k. Therefore, f_k is not necessarily strongly convex in general

and the optimal solution is not necessarily unique.

- We do not assume any condition for the rank of $E_k, k = 1, \ldots, K$ which is required to have full column rank in [47]. For the ADMM, this assumption is necessary to ensure that in each iteration, the subproblem for the k-th block is strongly convex. But for the iADA, this assumption is no longer required as there exists a proximal term in each subproblem which makes its optimality attainable and unique.

- The compactness assumption of $X_k, k = 1, \ldots, K$ will facilitate the proof in Section 2.4 and is not necessary for the convergence result in Section 2.5 due to the boundedness of the sequence generated by the iADA.

Based on these assumptions, we can simply write f as

$$f(x) = g(Ax) + h(x) = \sum_{k=1}^{K} g_k(A_k x_k) + \sum_{k=1}^{K} h_k(x_k) \tag{2.48}$$

where $g(Ax) = \sum_{k=1}^{K} g_k(A_k x_k)$ and $h(x) = \sum_{k=1}^{K} h_k(x_k)$ represent the smooth and nonsmooth parts respectively. In addition, $g(\cdot)$ is strongly convex and $h(\cdot)$ is convex with a polyhedral epigraph. The strong convexity of $g(\cdot)$ implies the following proposition, whose proof is omitted.

Proposition 2.1. *For any x in the solution set X^*, $A_k x_k, k = 1, \ldots, K$ are constant and hence Ax is constant.*

In the next section, we will discuss the stability result of the Lagrangian function under some perturbations which is essential to the local linear convergence result.

2.4 On the stability results of maximal monotone operator

In this section, we establish the stability result of the maximal monotone operator $\mathcal{T}_{\bar{L}}$ defined in (2.50) corresponding to the perturbations of both primal and dual solutions under Assumption 2.2. This property serves the key ingredient for the local convergence rate analysis of the iADA.

Recall the definition of $\bar{L}(w, x, \eta, \zeta)$ in (2.16). For each $(w, x, \eta, \zeta) \in W \times X \times S$, $\mathcal{T}_{\bar{L}}(w, x, \eta, \zeta)$ is defined as

$$\mathcal{T}_{\bar{L}}(w,x,\eta,\zeta) = \{(v_1,v_2,v_3,v_4)|(v_1,v_2,-v_3,-v_4) \in \partial \bar{L}(w,x,\eta,\zeta)\}, \tag{2.49}$$

or equivalently, $\mathcal{T}_{\bar{L}}(w,x,\eta,\zeta)$ is the set of $v = (v_1,v_2,v_3,v_4) \in (\mathbb{R}^m)^K \times \mathbb{R}^n \times (\mathbb{R}^m)^K \times (\mathbb{R}^m)^K$ such that

$$\begin{aligned} &\bar{L}(w',x',\eta,\zeta) - \langle w', v_1\rangle - \langle x', v_2\rangle + \langle \eta, v_3\rangle + \langle \zeta, v_4\rangle \\ \geq &\bar{L}(w,x,\eta,\zeta) - \langle w, v_1\rangle - \langle x, v_2\rangle + \langle \eta, v_3\rangle + \langle \zeta, v_4\rangle \\ \geq &\bar{L}(w,x,\eta',\zeta') - \langle w, v_1\rangle - \langle x, v_2\rangle + \langle \eta', v_3\rangle + \langle \zeta', v_4\rangle \\ &\text{for all } (w',x') \in W \times X, (\eta',\zeta') \in S. \end{aligned} \tag{2.50}$$

Any solution to $(0,0,0,0) \in \mathcal{T}_{\bar{L}}(w,x,\eta,\zeta)$ is a saddle point of $\bar{L}$. Denote $v_1 = (v_{1,1},\ldots,v_{1,K}) \in (\mathbb{R}^m)^K, v_2 = (v_{2,1},\ldots,v_{2,K}) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_K}$, $v_3 = (v_{3,1},\ldots,v_{3,K}) \in (\mathbb{R}^m)^K$ and $P_{W^\perp}(v_4) = (v_4^\perp,\ldots,v_4^\perp) \in (\mathbb{R}^m)^K$. We consider the following perturbed form of problem (2.11):

$$\begin{aligned} \min_{x,w} \quad & f_1(x_1) + \cdots + f_K(x_K) - \langle w, v_1\rangle - \langle x, v_2\rangle \\ \text{s.t.} \quad & E_k x_k - w_k + v_{3,k} + v_4^\perp = 0, \quad k = 1,\ldots,K-1, \\ & E_K x_K - q - w_K + v_{3,K} + v_4^\perp = 0, \\ & w_1 + \cdots + w_K = 0, \\ & x_k \in X_k, \quad k = 1,2,\ldots,K \end{aligned} \tag{2.51}$$

Its corresponding KKT conditions are given by

$$\begin{aligned} -E_k^T \eta_k + v_{2,k} \in \partial f_k(x_k) + N_{X_k}(x_k), \quad & k = 1,\ldots,K \\ -\eta_k + \mu = v_{1,k}, \quad & k = 1,\ldots,K \\ E_k x_k - w_k + v_{3,k} + v_4^\perp = 0, \quad & k = 1,\ldots,K-1 \\ E_K x_K - q - w_K + v_{3,K} + v_4^\perp = 0, & \\ w_1 + \cdots + w_K = 0. & \end{aligned} \tag{2.52}$$

One can easily check that

$$\begin{aligned}\mathcal{T}_{\bar{L}}^{-1}(v_1,v_2,v_3,v_4) = \ & \text{set of all } (w,x,\eta,P_{W^\perp}(\eta)) \in W\times X\times S \\ & \text{such that there exists } \mu\in\mathbb{R}^m \text{ satisfying that } (w,x,\eta,\mu) \\ & \text{is a solution of the KKT conditions (2.52).}\end{aligned} \tag{2.53}$$

Based on the above observation, we first study the stability results of the KKT system (2.52) under perturbations considered above. Under Assumption 2.2, every $f_k(x_k)$ is the sum of a smooth function $g_k(A_kx_k)$ and a nonsmooth function $h_k(x_k)$ with a polyhedral epigraph. By introducing a variable $s=(s_1,\ldots,s_K)\in\mathbb{R}^K$, for each k, we can rewrite the polyhedral set $\{(x_k,s_k): x_k\in X_k, h_k(x_k)\le s_k\}$ compactly as $C_x^kx_k+C_s^ks_k\ge c_k$ for some matrices $C_x^k\in\mathbb{R}^{j_k\times n_k}$, $C_s^k\in\mathbb{R}^{j_k\times 1}$ and $c_k\in\mathbb{R}^{j_k\times 1}$, where j_ks are some positive integers with $\sum_{k=1}^K j_k=j$. Then, we can transform (2.11) equivalently into

$$\begin{aligned}\min_{x,w,s}\quad & \sum_{k=1}^K g_k(A_kx_k)+s_k \\ \text{s.t.}\quad & E_kx_k-w_k=0,\quad k=1,\ldots,K-1, \\ & E_Kx_K-q-w_K=0, \\ & w_1+\cdots+w_K=0, \\ & C_x^kx_k+C_s^ks_k-c_k\ge 0,\quad k=1,2,\ldots,K.\end{aligned} \tag{2.54}$$

For the perturbed problem (2.51), similarly, we have the following equivalent transformation:

$$\begin{aligned}\min_{x,w,s}\quad & \sum_{k=1}^K g_k(A_kx_k)+s_k-\langle w,v_1\rangle-\langle x,v_2\rangle \\ \text{s.t.}\quad & E_kx_k-w_k+v_{3,k}+v_4^\perp=0,\quad k=1,\ldots,K-1, \\ & E_Kx_K-q-w_K+v_{3,K}+v_4^\perp=0, \\ & w_1+\cdots+w_K=0, \\ & C_x^kx_k+C_s^ks_k-c_k\ge 0,\quad k=1,2,\ldots,K.\end{aligned} \tag{2.55}$$

The canonical Lagrangian function for (2.55) is given by

$$
\begin{aligned}
L^v(w,x,s,\eta,\lambda,\mu) &= \sum_{k=1}^{K} g_k(A_k x_k) + s_k - \langle w, v_1\rangle - \langle x, v_2\rangle \\
&+ \sum_{k=1}^{K-1} \langle E_k x_k - w_k + v_{3,k} + v_4^{\perp}, \eta_k\rangle + \langle E_K x_K - q - w_K + v_{3,K} + v_4^{\perp}, \eta_K\rangle \\
&- \sum_{k=1}^{K} \langle C_x^k x_k + C_s^k s_k - c_k, \lambda_k\rangle + \langle w_1 + \cdots + w_K, \mu\rangle.
\end{aligned}
$$

We use $Sol(P(v_1, v_2, v_3, v_4))$ to denote the set of saddle points for the Lagrangian function $L^v(w, x, s, \eta, \lambda, \mu)$ defined above corresponding to the perturbed problem (2.55). Let $(v_1, v_2, v_3, v_4) = (0, 0, 0, 0)$, then $Sol(P(0, 0, 0, 0))$ represents the set of saddle points for the Lagrangian function of problem (2.54). In order to show the stability results for the KKT system (2.52), we define a set-valued mapping $\mathcal{M}$ that assigns the vector $(d, e, f) \in \mathbb{R}^n \times (\mathbb{R}^m)^K \times (\mathbb{R}^m)^K$ to the set of $(w, x, s, \eta, \lambda, \mu) \in (\mathbb{R}^m)^K \times \mathbb{R}^n \times \mathbb{R}^K \times (\mathbb{R}^m)^K \times \mathbb{R}^j \times \mathbb{R}^m$ that satisfy the following equations

$$
\begin{aligned}
-E_k^T \eta_k + (C_x^k)^T \lambda_k &= d_k, \quad k = 1, \ldots, K \\
-\eta_k + \mu &= e_k, \quad k = 1, \ldots, K \\
w_k - E_k x_k &= f_k, \quad k = 1, \ldots, K-1 \\
w_K - E_K x_K + q &= f_K, \\
w_1 + \cdots + w_K &= 0, \\
0 \le \lambda_k \perp C_x^k x_k + C_s^k s_k - c_k &\ge 0, \quad k = 1, \ldots, K \\
(C_s^k)^T \lambda_k &= 1, \quad k = 1, \ldots, K.
\end{aligned}
\tag{2.56}
$$

One can easily verify that

$$
\begin{aligned}
(w, x, s, \eta, \lambda, \mu) \in \mathcal{M}(A^T \nabla g(Ax) - v_2, v_1, v_3 + P_{W^\perp}(v_4)) \\
\text{if and only if } (w, x, s, \eta, \lambda, \mu) \in Sol(P(v_1, v_2, v_3, v_4)),
\end{aligned}
\tag{2.57}
$$

i.e., a solution of the KKT system of (2.55) is also a saddle point of the Lagrangian function (2.4).

By taking $(v_1, v_2, v_3, v_4) = (0, 0, 0, 0)$, we see that

$$(w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*) \in \mathcal{M}(A^T \nabla g(Ax^*), 0, 0)$$

if and only if

$$(w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*) \in Sol(P(0, 0, 0, 0)).$$

It is easily seen that $\mathcal{M}$ is a polyhedral multifunction; *i.e.*, the graph of $\mathcal{M}$ is the union of a finitely many polyhedral convex sets. In [78], Robinson established the following proposition that $\mathcal{M}$ enjoys the local upper Lipschitzian continuity property; see also [45].

Proposition 2.2. *There exists a positive scalar θ that depends on A, E, C_x, C_s only, such that for each $(\bar{d}, \bar{e}, \bar{f})$ there is a positive δ' satisfying*

$$\mathcal{M}(d, e, f) \subseteq \mathcal{M}(\bar{d}, \bar{e}, \bar{f}) + \theta ||(d, e, f) - (\bar{d}, \bar{e}, \bar{f})|| \mathcal{B} \text{ whenever } ||(d, e, f) - (\bar{d}, \bar{e}, \bar{f})|| \leq \delta' \qquad (2.58)$$

where $\mathcal{B}$ is the unit Euclidean ball in $(\mathbb{R}^m)^K \times \mathbb{R}^n \times \mathbb{R}^k \times (\mathbb{R}^m)^K \times \mathbb{R}^j \times \mathbb{R}^m$.

Based on this proposition, we claim that

Lemma 2.4. *Suppose Assumptions 2.1 and 2.2 hold. Then there exist positive scalars δ, τ depending on A, E, C_x, C_s only, such that for all $v = (v_1, v_2, v_3, v_4) \in (\mathbb{R}^m)^K \times \mathbb{R}^n \times (\mathbb{R}^m)^K \times (\mathbb{R}^m)^K$ and $||v|| \leq \delta$, any $(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) \in Sol(P(v_1, v_2, v_3, v_4))$, we have*

$$dist((w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)), Sol(P(0, 0, 0, 0))) \leq \tau ||v||. \qquad (2.59)$$

Proof. By the previous proposition, $\mathcal{M}$ is locally upper Lipschtizian with modulus θ at $(A^T \nabla g(Ax^*), 0, 0)$ for any $x^* \in X^*$. First we show that as $v \to 0$, $A^T \nabla g(Ax(v)) \to A^T \nabla g(Ax^*)$. For that, take a sequence $v^i = (v_1^i, v_2^i, v_3^i, v_4^i) \in (\mathbb{R}^m)^K \times \mathbb{R}^n \times (\mathbb{R}^m)^K \times (\mathbb{R}^m)^K, i = 1, 2, \cdots$, such that $||v^i|| \to 0$. Based on Assumption 2.2(e), the sequence $x(v^i), i = 1, 2, \cdots$ lies in a compact set and so the other sequence $s(v^i)$ and $w(v^i)$ also belong to some compact sets, given the fact $s(v^i) = h(x(v^i))$ and the linear relationship among $x(v^i), v^i$ and $w(v^i)$. By passing to a subsequence if necessary, let $(w^\infty, x^\infty, s^\infty)$ be a cluster point of $\{(w(v^i), x(v^i), s(v^i)\}$. Due to the continuity of $\nabla g(\cdot)$,

$(A^T\nabla g(Ax(v^i)) - v_2^i, v_1^i, v_3^i + P_{W^\perp}(v_4^i))$ converges to $(A^T\nabla g(Ax^\infty), 0, 0)$ as $i \to \infty$. For all i, $\{(w(v^i), x(v^i), s(v^i), A^T\nabla g(Ax(v^i)) - v_2^i, v_1^i, v_3^i + P_{W^\perp}(v_4^i))\}$ lies in the set

$$\{(w, x, s, d, e, f) | (w, x, s, \eta, \lambda, \mu) \in \mathcal{M}(d, e, f) \text{ for some } (\eta, \lambda, \mu)\}$$

which is a closed polyhedral set. By passing to the limit, we can conclude

$$(w^\infty, x^\infty, s^\infty, \eta^\infty, \lambda^\infty, \mu^\infty) \in \mathcal{M}(A^T\nabla g(Ax^\infty), 0, 0)$$

for some $(\eta^\infty, \lambda^\infty, \mu^\infty) \in (\mathbb{R}^m)^K \times \mathbb{R}^j \times \mathbb{R}^m$. From Proposition 2.1, we know $Ax^\infty = Ax^*$ for any $x^* \in X^*$ which further implies that $A^T\nabla g(Ax(v)) \to A^T\nabla g(Ax^*)$. Then there exists a positive scalar δ such that for all v satisfying $||v|| \leq \delta$, the following inequality

$$||A^T\nabla g(Ax(v)) - A^T\nabla g(Ax^*)|| + ||v|| \leq \delta'$$

holds. Based on Proposition 2.2, there exists $(w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*) \in \mathcal{M}(A^T\nabla g(Ax^*), 0, 0)$, satisfying

$$\begin{aligned} &||(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) - (w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*)|| \\ &\qquad \leq \theta(||A^T\nabla g(Ax(v)) - A^T\nabla g(Ax^*)|| + ||v||). \end{aligned}$$

Since $(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) \in \mathcal{M}(A^T\nabla g(Ax) - v_2, v_1, v_3 + P_{W^\perp}(v_4))$, by the definition of $\mathcal{M}$ we have

$$\begin{aligned} -E_k^T\eta_k(v) + (C_x^k)^T\lambda_k(v) &= A_k^T\nabla g_k(A_k x_k(v)) - v_2, \quad k = 1, \ldots, K \\ -\eta_k(v) + \mu(v) &= v_{1,k}, \quad k = 1, \ldots, K \\ w_k(v) - E_k x_k(v) &= v_{3,k} + v_4^\perp, \quad k = 1, \ldots, K-1 \\ w_K(v) - E_K x_K(v) + q &= v_{3,K} + v_4^\perp, \\ w_1(v) + \cdots + w_K(v) &= 0, \\ 0 \leq \lambda_k(v) \perp C_x^k x_k(v) + C_s^k s_k(v) - c_k &\geq 0, \quad k = 1, \ldots, K \\ (C_s^k)^T\lambda_k(v) &= 1, \quad k = 1, \ldots, K. \end{aligned} \tag{2.60}$$

Similarly, since $(w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*) \in \mathcal{M}(A^T \nabla g(Ax^*), 0, 0)$, it follows that

$$\begin{aligned}
-E_k^T \eta_k^* + (C_x^k)^T \lambda_k^* &= A_k^T \nabla g_k(A_k x_k^*), \quad k = 1, \ldots, K \\
-\eta_k^* + \mu^* &= 0, \quad k = 1, \ldots, K \\
w_k^* - E_k x_k^* &= 0, \quad k = 1, \ldots, K-1 \\
w_K^* - E_K x_K^* + q &= 0, \\
w_1^* + \cdots + w_K^* &= 0, \\
0 \leq \lambda_k^* \perp C_x^k x_k^* + C_s^k s_k^* - c_k &\geq 0, \quad k = 1, \ldots, K \\
(C_s^k)^T \lambda_k^* &= 1, \quad k = 1, \ldots, K.
\end{aligned} \tag{2.61}$$

Due to the strong convexity of $g_k(\cdot)$ and the Lipschitzian continuity of its derivative $\nabla g_k(\cdot)$ in Assumption 2.2, there exist positive scalars σ_g^k, L_g^k such that for all $x_1^k, x_2^k \in X_k$

$$\langle A_k^T \nabla g_k(A_k x_1^k) - A_k^T \nabla g_k(A_k x_2^k), x_1^k - x_2^k \rangle \geq \sigma_g^k ||A_k x_1^k - A_k x_2^k||^2,$$

and

$$||A_k^T \nabla g_k(A_k x_1^k) - A_k^T \nabla g_k(A_k x_2^k)|| \leq L_g^k ||A_k x_1^k - A_k x_2^k||.$$

Define $\sigma_g = \min_k \sigma_g^k$ and $L_g = \max_k L_g^k$. Taking $x_1 = x(v), x_2 = x^*$, we obtain

$$\begin{aligned}
&\sigma_g \sum_{k=1}^K ||A_k(x(v)_k - x_k^*)||^2 \\
\leq &\sum_{k=1}^K \langle A_k^T \nabla g_k(A_k x(v)_k) - A_k^T \nabla g_k(A_k x_k^*), x(v)_k - x_k^* \rangle \\
= &\sum_{k=1}^K \langle -E_k^T(\eta(v)_k - \eta_k^*) + (C_x^k)^T(\lambda(v)_k - \lambda_k^*) + v_{2,k}, x(v)_k - x_k^* \rangle \\
= &\sum_{k=1}^K \langle \lambda(v)_k - \lambda_k^*, C_x^k x(v)_k - C_x^k x_k^* \rangle + \sum_{k=1}^K \langle \eta(v)_k - \eta_k^*, -E_k x(v)_k + E_k x_k^* \rangle \\
&+ \sum_{k=1}^K \langle v_{2,k}, x(v)_k - x_k^* \rangle
\end{aligned}$$

where the first inequality comes from (2.4) and the equalities come from (2.60) and (2.61). Moreover,

we have

$$\begin{aligned}
&\sum_{k=1}^{K}\langle\lambda(v)_k-\lambda_k^*, C_x^k x(v)_k - C_x^k x_k^*\rangle \\
&=\sum_{k=1}^{K}\langle\lambda(v)_k-\lambda_k^*, C_x^k x(v)_k - C_x^k x_k^*\rangle + \langle\sum_{k=1}^{K}\lambda(v)_k-\lambda_k^*, C_s^k s(v)_k - C_s^k s_k^*\rangle \\
&=\sum_{k=1}^{K}\langle\lambda(v)_k-\lambda_k^*, (C_x^k x(v)_k + C_s^k s(v)_k - c_k) - (C_x^k x_k^* + C_s^k s_k^* - c_k)\rangle \\
&=-\sum_{k=1}^{K}[\langle\lambda_k^*, C_x^k x(v)_k + C_s^k s(v)_k - c_k\rangle + \langle\lambda(v)_k, C_x^k x_k^* + C_s^k s_k^* - c_k\rangle] \le 0
\end{aligned}$$

where the first equality follows from the fact that $(C_s^k)^T\lambda(v)_k = (C_s^k)^T\lambda_k^* = 1, k = 1,\ldots,K$ and the last equality and inequality both result from the complementary conditions in (2.60) and (2.61). Consequently, we obtain that

$$\begin{aligned}
&\sigma_g\sum_{k=1}^{K}||A_k(x(v)_k - x_k^*)||^2 \\
&\le\sum_{k=1}^{K}\langle\eta(v)_k-\eta_k^*, -E_k x(v)_k + E_k x_k^*\rangle + \sum_{k=1}^{K}\langle v_{2,k}, x(v)_k - x_k^*\rangle \\
&=\sum_{k=1}^{K}\langle\mu(v)+v_{1,k}-\mu^*, -w(v)_k + w_k^* + v_{3,k} + v_4^{\perp}\rangle + \sum_{k=1}^{K}\langle v_{2,k}, x(v)_k - x_k^*\rangle \\
&=\sum_{k=1}^{K}\langle\mu(v)-\mu^*, v_{3,k}+v_4^{\perp}\rangle + \underbrace{\sum_{k=1}^{K}\langle\mu(v)-\mu^*, -w(v)_k + w_k^*\rangle}_{=0} \\
&+\sum_{k=1}^{K}\langle v_{1,k}, -w(v)_k + w_k^* + v_{3,k} + v_4^{\perp}\rangle + \sum_{k=1}^{K}\langle v_{2,k}, x(v)_k - x_k^*\rangle \\
&\le||\mu(v)-\mu^*||(||v_3|| + ||v_4||) + ||w(v)-w^*||||v_1|| + ||v_1||(||v_3|| + ||v_4||) + ||(x(v)-x^*)||||v_2|| \\
&\le||(w(v), x(v), \mu(v)) - (w^*, x^*, \mu^*)||||v|| + ||v||^2.
\end{aligned}$$

Finally, based on Proposition 2.2 and the above inequality, we have

$$
\begin{aligned}
&||(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) - (w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*)||^2 \\
&\leq \theta^2(||A^T\nabla g(Ax(v)) - A^T\nabla g(Ax^*)|| + ||v||)^2 \\
&\leq 2\theta^2(\sum_{k=1}^{K} ||A_k^T\nabla g_k(A_k x(v)_k) - A_k^T\nabla g_k(A_k x_k^*)||^2 + ||v||^2) \\
&\leq 2\theta^2(L_g^2\sum_{k=1}^{K} ||A_k(x(v)_k - x_k^*)||^2 + ||v||^2) \\
&\leq 2\theta^2\max\{\frac{L_g^2}{\sigma_g}, 1\}(\sigma_g\sum_{k=1}^{K} ||A_k(x(v)_k - x_k^*)||^2 + ||v||^2) \\
&\leq 2\theta^2\max\{\frac{L_g^2}{\sigma_g}, 1\}(||(w(v), x(v), \mu(v)) - (w^*, x^*, \mu^*)||||v|| + 2||v||^2) \\
&\leq 2\theta^2\max\{\frac{L_g^2}{\sigma_g}, 1\}(||(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) - (w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*)||||v|| + 2||v||^2)
\end{aligned}
$$

We see the above inequality is quadratic in

$$||(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) - (w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*)||/||v||,$$

so we have

$$||(w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)) - (w^*, x^*, s^*, \eta^*, \lambda^*, \mu^*)||/||v|| \leq \tau$$

for some scalar τ depending on θ, L_g, σ_g only. We conclude that

$$dist((w(v), x(v), s(v), \eta(v), \lambda(v), \mu(v)), Sol(P(0,0,0,0))) \leq \tau||v||.$$

□

In view of the operator $\mathcal{T}_{\bar{L}}$, combining Lemma 2.4 with the observation in (2.53), we have the following corollary.

Corollary 2.1. *Suppose Assumptions 2.1 and 2.2 hold. Then there exist positive scalars δ, τ depending on A, E, C_x, C_s only, such that for all $v = (v_1, v_2, v_3, v_4) \in (\mathbb{R}^m)^K \times \mathbb{R}^n \times (\mathbb{R}^m)^K \times (\mathbb{R}^m)^K$*

and $||v|| \leq \delta$, *any* $(w(v), x(v), \eta(v), \zeta(v)) \in \mathcal{T}_{\bar{L}}^{-1}(v)$ *satisfies*

$$dist((w(v), x(v), \eta(v), \zeta(v)), \mathcal{T}_{\bar{L}}^{-1}(0,0,0,0)) \leq 2\tau||v||. \tag{2.62}$$

Proof. From Lemma 2.4 and observation in (2.53) , we know that for any $(w(v), x(v), \eta(v), \zeta(v)) \in \mathcal{T}_{\bar{L}}^{-1}(v)$, there exists a $(w^*, x^*, \eta^*, \zeta^*) \in \mathcal{T}_{\bar{L}}^{-1}(0,0,0,0)$ satisfying that

$$||(w(v), x(v), \eta(v)) - (w^*, x^*, \eta^*)|| \leq \tau||v||.$$

Since $\zeta(v) = P_{W^\perp}(\eta(v))$ and $\zeta^* = P_{W^\perp}(\eta^*)$, then

$$||(w(v), x(v), \eta(v), \zeta(v)) - (w^*, x^*, \eta^*, \zeta^*)|| \leq 2\tau||v||$$

holds which leads to (2.62). □

The compactness assumption of X_k is indeed necessary for Corollary 2.1. However, if the generated sequence $\{x(v^i)\}$ lies in a compact set for a sequence $\{v^i\}_{i=1}^{\infty}$ converging to the origin, we claim the following result: under Assumptions 2.1 and 2.2(a)-(d), there exist positive scalars δ, τ depending on A, E, C_x, C_s only, when $||v^i|| \leq \delta$ the following

$$dist((w(v^i), x(v^i), \eta(v^i), \zeta(v^i)), \mathcal{T}_{\bar{L}}^{-1}(0,0,0,0)) \leq 2\tau||v^i||$$

holds. This observation relaxes the compactness assumption for $X_k, k = 1, \ldots, K$ (Assumption 2.2(e)) when we show the local linear convergence in Theorem 2.5 for the iADA in Section 2.5.

2.5 Convergence analysis of the inexact ADA

In this section, we study the convergence results of the inexact ADA for solving the problem (2.1). For that, we first need to adopt the following stopping criterion developed in [80, 81] for approximately solving these subproblems

$$\text{dist}(0, \partial\phi_{k,\rho,c}^{\nu}(x_k^{\nu+1})) \leq \frac{\epsilon_\nu}{cK(\rho||E|| + ||E|| + 1)}, \qquad \sum_{\nu=0}^{\infty} \epsilon_\nu < \infty. \tag{A}$$

Theorem 2.4. *Suppose Assumption 2.1 holds and let* $\{(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)\}$ *in* $W \times X \times S$ *be the infinite sequence generated by the ADA with the stopping criterion* (A). *Then* $(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ *converges to some saddle point* $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ *of* (2.16) *such that*

(a) $(\bar{w}, \bar{x})$ *solves* (2.11), *hence* $\bar{x}$ *solves* (2.1),

(b) $\bar{\eta}_1 = \cdots = \bar{\eta}_K \in \mathbb{R}^m$, *and this common multiplier vector solves* (2.10).

Proof. In each iteration ν, we denote $(w_0^{\nu+1}, x_0^{\nu+1}, \eta_0^{\nu+1}, \zeta_0^{\nu+1}) = P_\nu(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ as the exact saddle point of $\bar{L}^\nu(w, x, \eta, \zeta)$ and $(w^{\nu+1}, x^{\nu+1}, \eta^{\nu+1}, \zeta^{\nu+1})$ as the inexact saddle point generated following the stopping criteria (A) respectively. By the update rule, the following estimates hold:

$$||\eta_0^{\nu+1} - \eta^{\nu+1}|| \leq \frac{\rho||E||}{2}||x_0^{\nu+1} - x^{\nu+1}||,$$

$$||\zeta_0^{\nu+1} - \zeta^{\nu+1}|| \leq \frac{\rho||E||}{2}||x_0^{\nu+1} - x^{\nu+1}||,$$

and

$$||w_0^{\nu+1} - w^{\nu+1}|| \leq ||E||||x_0^{\nu+1} - x^{\nu+1}||.$$

Thus, we can obtain

$$||(w^{\nu+1}, x^{\nu+1}, \eta^{\nu+1}, \zeta^{\nu+1}) - P_\nu(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)|| \leq (\rho||E|| + ||E|| + 1)||x^{\nu+1} - x_0^{\nu+1}||. \tag{2.63}$$

Observing that the function $\phi_{k,\rho,c}^\nu$ defined in (2.20) is strongly convex with modulus at least $\frac{1}{c}$ and $x_{0,k}^{\nu+1}$ minimize $\phi_{k,\rho,c}^\nu(x_k)$, we get

$$||x^{\nu+1} - x_0^{\nu+1}|| \leq c\sum_{k=1}^{K} \text{dist}(0, \partial\phi_{k,\rho,c}^{\nu+1}(x_k^{\nu+1})). \tag{2.64}$$

Combining criterion (A), (2.63) and (2.64), we have

$$||(w^{\nu+1}, x^{\nu+1}, \eta^{\nu+1}, \zeta^{\nu+1}) - P_\nu(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)|| \leq \epsilon_\nu, \text{ with } \sum_{\nu=1}^{\infty} \epsilon_\nu < \infty. \tag{2.65}$$

From Assumption 2.1, there exists a saddle point of the Lagrangian (2.2). Therefore based on the

relationship between (2.2) and (2.16) in Lemma 2.1, there exists at least one saddle point of the Lagrangian function $\bar{L}$. On the basis of [81], the sequence of elements $(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ generated in this manner from any initial $(w^1, x^1) \in W \times X$ and $(\eta^1, \zeta^1) \in S$ converges to some saddle point $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ of the $\bar{L}$. Then $(\bar{w}, \bar{x})$ solves (2.11) and $(\bar{\eta}, \bar{\zeta})$ solves (2.17). By Lemma 2.1, both (a) and (b) hold. □

For the local convergence analysis, we need the following stopping criteria

$$\text{dist}(0, \partial\phi^\nu_{k,\rho,c}(x_k^{\nu+1})) \leq \frac{\epsilon'_\nu}{cK(\rho||E|| + ||E|| + 1)} \min\{1, ||x_k^{\nu+1} - x_k^\nu||\}, \qquad \sum_{\nu=0}^{\infty} \epsilon'_\nu < \infty. \tag{B}$$

The iADA does not impose any condition on the choice of c. We set $c = \rho$ for simplicity of the following analysis. The coefficient $\rho/2$ for the primal proximal term $||w - w^\nu||^2$ in (2.19) can be changed to $1/2\rho$ after the rescaling $w' = \rho w$ and such rescaling only applies to the magnitude of w and does not bring any other changes to the iADA. So this distinction from the standard proximal point method for minimax problems in [80, Section 5] will not influence the following convergence results.

Theorem 2.5. *Suppose Assumptions 2.1 and 2.2 hold and let* $\{(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)\}$ *in* $W \times X \times S$ *be the infinite sequence generated by the ADA with the stopping criterion* (B)*. Then,* $(w^\nu, x^\nu, \eta^\nu, \zeta^\nu)$ *converges to some saddle point* $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ *of* (2.16) *and there exists* $\{\theta_\nu\}$ *such that*

$$dist((w^{\nu+1}, x^{\nu+1}, \eta^{\nu+1}, \zeta^{\nu+1}), \mathcal{T}_{\bar{L}}^{-1}((0,0,0,0))) \leq \theta_\nu dist((w^\nu, x^\nu, \eta^\nu, \zeta^\nu), \mathcal{T}_{\bar{L}}^{-1}((0,0,0,0)))$$

for sufficient large ν *and* $\lim_{\nu\to\infty} \theta_\nu = \frac{2\tau}{\sqrt{(4\tau^2+\rho^2)}} < 1$ *for some* τ*.*

Proof. From Corollary 2.62, we have shown that there exist $\tau, \delta > 0$ such that for all $v = (v_1, v_2, v_3, v_4) \in (\mathbb{R}^m)^K \times \mathbb{R}^n \times (\mathbb{R}^m)^K \times (\mathbb{R}^m)^K$ and $||v|| \leq \delta$, any $(w(v), x(v), \eta(v), \zeta(v)) \in \mathcal{T}_{\bar{L}}^{-1}(v)$ satisfies

$$dist((w(v), x(v), \eta(v), \zeta(v)), \mathcal{T}_{\bar{L}}^{-1}((0,0,0,0))) \leq 2\tau||v||. \tag{2.66}$$

So this theorem follows from [64, Theorem 2.1]. □

Remark 1. In Theorem 2.4, we have shown that the sequence $\{u^\nu = (w^\nu, x^\nu, \eta^\nu, \zeta^\nu)\}$ converges

to some saddle point $(\bar{w}, \bar{x}, \bar{\eta}, \bar{\zeta})$ of (2.16) and hence $\{u^\nu\}$ lies in a compact set. Based on the observation in (2.4) and the proof of [64, Theorem 2.1], the compactness of assumption of X_k (Assumption 2.2(e)) is no longer needed for Theorem 2.5.

Remark 2. When $c \neq \rho$, the local linear convergence still holds while the convergence rate ($\lim_{\nu\to\infty} \theta_\nu$) changes.

Next, we provide some well-known examples on which the iADA enjoys the local linear convergence.

Convex regularization. Many problems from empirical risk minimization and variable selection can be written as the following:

$$\min_x f(x; (A, b)) + r(x) \tag{2.67}$$

where $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, $f(\cdot)$ is the loss function which is often strongly convex with Lipschitz continuous gradient and $r(\cdot)$ is a convex regularization term which is possibly nonsmooth (*e.g.*, the ℓ_1-norm and TV-norm). By adding the constraint $x - z = 0$, the above problem can be reformulate as

$$\begin{aligned} &\min_{x,z} f(x; (A, b)) + r(z) \\ &\text{s.t.} \quad x - z = 0. \end{aligned} \tag{2.68}$$

Exchange problem. Consider a network with K agents exchanging n commodities. Let $x_k \in \mathbb{R}^n$ be the amount of commodities in each agent k and $f_k : \mathbb{R}^n \to \mathbb{R}$ be its corresponding cost function. The exchange problem is given by

$$\min_{\{x_k\}_{k=1}^K} \sum_{k=1}^K f_k(x_k) \qquad \text{s.t.} \ \sum_{k=1}^K x_k = 0$$

which minimizes the total cost subject to the equilibrium constraint on all K agents. In this special case, $E_k = I$ and $q = 0$. Optimization problems in this form arise in many areas such as resource allocation [6, 108], multi-agent system [112] and image processing [106]. When the cost function f_k in each agent satisfies Assumption 2.2(a)-(c), based on Theorem 2.5, local linear convergence result is valid for the iADA under certain approximation criteria.

2.6 Numerical Examples

In this section, we demonstrate the linear convergence of both the exact ADA and the inexact ADA by some simple numerical examples. All the computational tasks for numerical experiments are implemented in Matlab 2017b running on a MacBook Pro. Retina, 2.6 GHz Intel Core i7 with 16Gb 2133 MHz LPDDR3 memory.

2.6.1 The *lasso* problem

We perform some numerical experiments of Algorithm 2.1 for solving the following *lasso* problem:

$$\min_{x\in\mathbb{R}^d} \frac{1}{2}||Ax-b||_2^2 + \lambda_1||x||_1 \tag{2.69}$$

where $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$ and λ_1 is the regularization parameter. By introducing an auxiliary variable $z \in \mathbb{R}^d$, the above problem is equivalent to

$$\begin{aligned} &\min_{x,z\in\mathbb{R}^d} \frac{1}{2}||Ax-b||_2^2 + \lambda_1||z||_1 \\ &\text{s.t.} \quad x - z = 0. \end{aligned} \tag{2.70}$$

Clearly, (2.70) is a two-block decomposition problem with $f_1(x_1) = \frac{1}{2}||Ax_1-b||_2^2$ and $f_2(x_2) = \lambda_1||x_2||_1$ by replacing x and z with x_1 and x_2. Notice that f_1 and f_2 are not necessarily strongly convex. In this case,

$$\begin{aligned} \phi^\nu_{1,\rho,c}(x_1) &= \frac{1}{2}||Ax_1-b||_2^2 + \frac{\rho}{4}||x_1 - w_1^\nu + \frac{2}{\rho}y_1^\nu||_2^2 + \frac{1}{2c}||x_1-x_1^\nu||_2^2, \\ \phi^\nu_{2,\rho,c}(x_2) &= \lambda_1||x_2||_1 + \frac{\rho}{4}||x_2 + w_2^\nu - \frac{2}{\rho}y_2^\nu||_2^2 + \frac{1}{2c}||x_2-x_2^\nu||_2^2. \end{aligned} \tag{2.71}$$

For the first block, we can derive that

$$x_1^{\nu+1} = [A^TA + (\frac{\rho}{2}+\frac{1}{c})\mathbf{I}_d]^{-1}(A^Tb + \frac{\rho}{2}w_1^\nu + \frac{x_1^\nu}{c} - y_1^\nu). \tag{2.72}$$

Though it may be time consuming to compute $[A^TA+(\frac{\rho}{2}+\frac{1}{c})\mathbf{I}_d]^{-1}$ when d is large, we only need to compute it at the initialization stage. The special structure of $A^TA+(\frac{\rho}{2}+\frac{1}{c})\mathbf{I}_d$ can be exploited and substantially improve performance, see [11, Section 4.2]. For the second block, the exact solution

to the subproblem in each iteration is given by

$$x_2^{\nu+1} := S(\frac{y_2^\nu + x_2^\nu/c - \rho w_2^\nu/2}{\rho/2 + 1/c}, \frac{\lambda_1}{\rho/2 + 1/c}) \tag{2.73}$$

where the *soft thresholding operator* S is defined in [11].

We generate the matrix A and $0.05d$ nonzero entries of the sparse vector $x_0 \in \mathbb{R}^d$ from the standard Gaussian distribution $\mathcal{N}(0,1)$. We then let the response vector $b \in \mathbb{R}^n$ be given by $b = Ax_0 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 10^{-3}\mathbf{I}_n)$ and let the regularization parameter λ_1 be $0.1||A^T b||_\infty$. We test the algorithm on two different sets of (n,d): $(1000, 4000)$, $(2000, 20000)$.

In our test, we compare the result of ADA with two other methods for the *lasso* problem: ADMM[11] and P-PPA[5]. For the implementation of ADMM, we take a widely-used step-length 1.618 and a fixed penalty parameter 1. For P-PPA, we used the parameters suggested in [5] for solving the *lasso*. For the ADA, we choose the following three pairs of (ρ, c): $(1,1), (5,5), (10,10)$. In each iteration, we solve both subproblems exactly and the computational time for all three algorithms is nearly the same. For all algorithms, we use the same initial point $(x^0, y^0) = (\mathbf{0}, \mathbf{0})$ and run 300 iterations. For all comparison algorithms, we report the objective value $f(x^\nu) = \frac{1}{2}||Ax_1^\nu - b||^2 + \lambda_1||x_2^\nu||_1$, and the residual norm $||x_1^\nu - x_2^\nu||$. The convergence results are presented in Figures 2.1 and 2.2. From Figure 2.1, we notice that ADMM performs best in the case

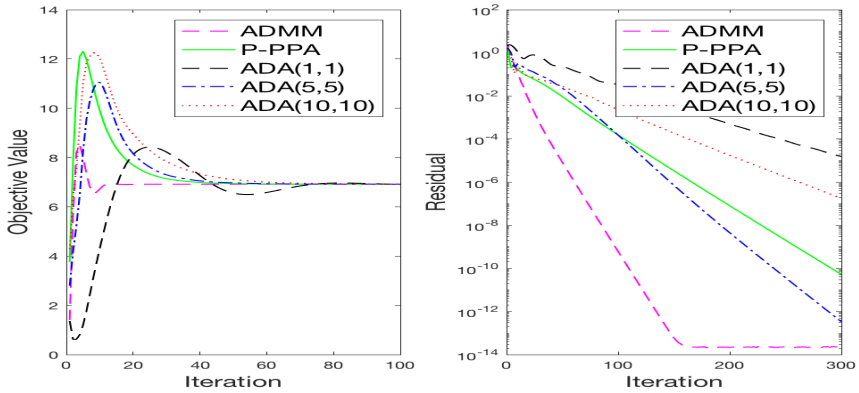


Figure 2.1: Convergence results of ADA, ADMM and P-PPA for the *lasso*: $(n,d) = (1000, 4000)$.

$(n,d) = (1000, 4000)$ while ADA achieves comparable performance with P-PPA when $(\rho, c) = (5,5)$. This suggests that the convergence of ADA becomes slow if the proximal parameter is either too

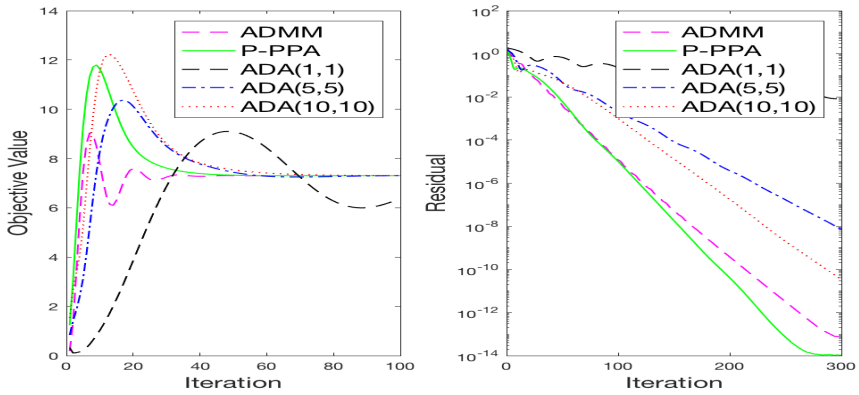


Figure 2.2: Convergence results of ADA, ADMM and P-PPA for the *lasso*: $(n, d) = (2000, 20000)$.

big or too small. When $(n, d) = (2000, 20000)$, P-PPA shows the best convergence and ADA with $(\rho, c) = (10, 10)$ converges a little bit slower. Both ADMM and P-PPA methods use the Gauss-Seidel style update which tends to converge faster in terms of iterations, since it is able to incorporate information from the other coordinates more quickly. However, the Jacobi style update of ADA is more amenable for parallelization.

2.6.2 The exchange problem

For the exchange problem in (2.5), we consider the quadratic cost function $f_k(x_k) = \frac{1}{2}||A_k x_k - b_k||^2$ where $A_k \in \mathbb{R}^{p\times n}$ and $b_k \in \mathbb{R}^p$, $k = 1, \ldots, K$. Then, the subproblems in each iteration can be written as

$$x_k^{\nu+1} = \underset{x_k}{\operatorname{argmin}} \frac{1}{2}||A_k x_k - b_k||^2 + r||x_k - d_k^\nu||^2, \quad \forall k = 1, \ldots, K,$$

for some $r \in \mathbb{R}_+$ and $d_k^\nu \in \mathbb{R}^n$. Notice that the matrices $A_k^T A_k + 2r\mathbf{I}_n, k = 1, \ldots, K$ are positive definite since $r > 0$. We only have to compute $(A_k^T A_k + 2r\mathbf{I}_n)^{-1}$ for one time before the iterations start. In the experiments, we randomly generate the optimal solution $x_k^*, k = 1, \ldots, K-1$ by the standard normal distribution and set $x_K^* = -\sum_{k=1}^{K-1} x_k^*$. The matrices $A_k, k = 1, \ldots, K$ are generated from standard Gaussian distribution and we let $b_k = A_k x_k^*$. In this setting, x^* is an optimal solution to (2.5) but not necessarily the unique one, and the optimal value is 0. We set $K = 20$, $n = 1000$, $p = 800$, and none of $f_k(x_k), k = 1, \ldots, K$ is strongly convex. We compare

the performance of ADA with VSADMM and Prox-JADMM mentioned in Section 2.2.4. For the implementation of VSADMM and Prox-JADMM, we use codes provided in [23]. For the proximal parameters of ADA, we set $(\rho, c) = (10, 10)$ in the experiment.

For all of the algorithms, we start from the same initial point $(x^0, y^0) = (\mathbf{0}, \mathbf{0})$ and run 500 iterations. Figure 2.3 shows the objective function value $\sum_{k=1}^{K} f_k(x_k)$ and the residual $||\sum_{k=1}^{K} x_k||$ of each iteration for the average outcome of 10 random simulations. We can see that ADA shows a better convergence of the objective value compared with VSADMM and is slower than Prox-JADMM in terms of iterations. However, Prox-JADMM requires extra computational time to update the proximal parameters which is shown in Figure 2.4. Overall, ADA shows competitive convergence results in this experiment compared with two variants of the classical ADMM method which facilitate parallelization.

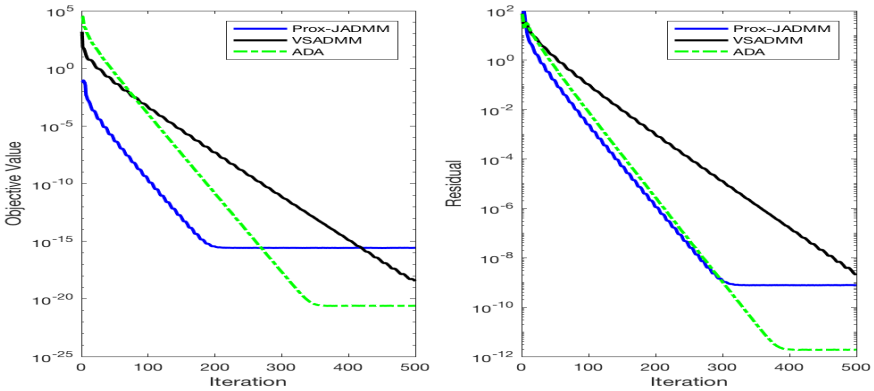


Figure 2.3: Exchange Problem: $K = 20, n = 1000, p = 800$. Convergence results versus iteration.

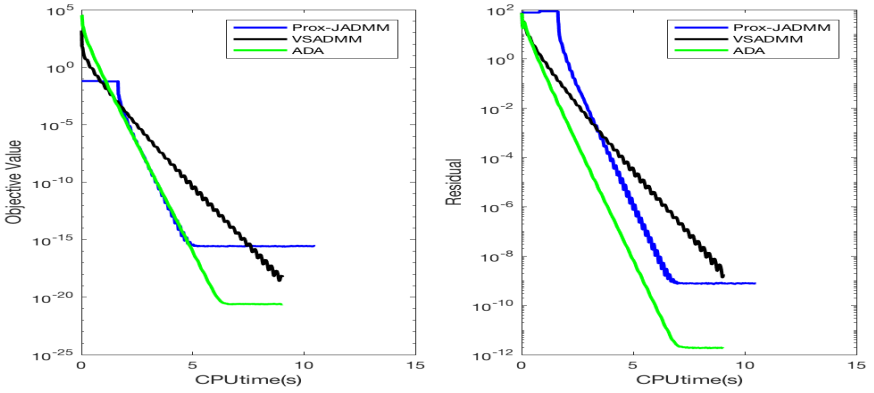


Figure 2.4: Exchange Problem: $K = 20, n = 1000, p = 800$. Convergence results versus time.

2.6.3 Distributed sparse logistic regression

Here, we use iADA to solve the convex regularization problem (2.68) with a modest number of features but a relative large number of training examples. Many statistical problems belong to this regime, with a large n and a small d dataset. In particular, we consider the following ℓ_1-regularized logistic regression:

$$\min_{x\in\mathbb{R}^d} F(x) = \sum_{j=1}^{n} \ell(x;(a_j,b_j)) + \lambda||x||_1 \tag{2.74}$$

where $(a_j,b_j)\in\mathbb{R}^{d+1}, j=1,\ldots,n$ and $\ell(x;(a_j,b_j)) = \log(1+\exp(-b_j a_j^T x))$. For the purpose of parallel computation, we partition $A\in\mathbb{R}^{n\times d}$ and $b\in\mathbb{R}^n$ into N blocks

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_N \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b^1 \\ \vdots \\ b^N \end{bmatrix},$$

with $A_i\in\mathbb{R}^{n_i\times d}$ and $b^i\in\mathbb{R}^{n_i}$. Define $\bar{n}_i = \sum_{j=1}^{i} n_j$ and we notice $\bar{n}_0 = 0$ and $\bar{n}_N = \sum_{j=1}^{N} n_j = n$. By introducing variables $x_i\in\mathbb{R}^d, i=1,\ldots,N$, (2.74) can be transformed into the following:

$$\begin{aligned} &\min_{x_i,z\in\mathbb{R}^d} \sum_{i=1}^{N} \ell_i(x_i;(A_i,b^i)) + \lambda||z||_1 \\ &\text{s.t.} \quad x_i - z = 0, \quad i=1,\ldots,N. \end{aligned} \tag{2.75}$$

where $\ell_i(x_i;(A_i,b^i)) = \sum_{j=\bar{n}_{i-1}+1}^{\bar{n}_i} \log(1+\exp(-b_j a_j^T x_i))$. In our experiment, we use two publicly available datasets: (1) the `w8a` dataset (49749 examples and 300 features) and (2) the `ijcnn1` dataset (49990 examples and 22 feature). The main step of iADA algorithm is given by

$$\begin{aligned} x_i^{\nu+1} &\approx \underset{x_i}{\operatorname{argmin}} \underbrace{\ell_i(x_i;(A_i,b^i)) + \frac{\rho}{4}||x_i - w_{x,i}^{\nu} + \frac{2}{\rho}y_{x,i}^{\nu}||_2^2 + \frac{1}{2c}||x_i - x_i^{\nu}||_2^2}_{\phi_{i,\rho,c}^{\nu}(x_i)}, \\ z^{\nu+1} &= \underset{z}{\operatorname{argmin}}\, \lambda_1||z||_1 + \frac{\rho}{4}\sum_{i=1}^{N}||z + w_{z,i}^{\nu} - \frac{2}{\rho}y_{z,i}^{\nu}||_2^2 + \frac{1}{2c}||z - z^{\nu}||_2^2, \end{aligned} \tag{2.76}$$

where $w_x^\nu = (w_{x,1}^\nu, \ldots, w_{x,N}^\nu) \in \mathbb{R}^{Nd}, y_x^\nu = (y_{x,1}^\nu, \ldots, y_{x,N}^\nu) \in \mathbb{R}^{Nd}$ and $w_z^\nu, y_z^\nu \in \mathbb{R}^{Nd}$. The x_i update involves an ℓ_2 regularized logistic regression which cannot be solved exactly. Here, we use the L-BFGS algorithm to solve them until the inexact criteria (A) and (B) are satisfied. Such criteria can be checked by identifying the norm of the gradient $||\nabla \phi_{i,\rho,c}^\nu(x_i)||$. For the z update, exact solutions can be derived by the soft threshold operator.

For comparison, we consider the inexact ADMM (iADMM) method proposed in [27, 39]. Similar subproblems as (2.76) will arise for x_i and z updates. An analogous inexact criterion as (A) are proposed in [27, 39] to guarantee the convergence of the inexact ADMM and can also be verified by examining the norm of the gradient in the x_i updates.

In the experiment, we set $\epsilon_\nu = \frac{1}{\nu^\gamma}$ with $\gamma = 1.0, 1.5, 2.0$ to control the inexactness of the x_i updates in both algorithms. We also consider different partitions with $N = 20, 50$. For the implementation of iADMM, we use a step-length 1.618 and a fixed penalty parameter 10 after tuning. For iADA, we choose the proximal parameters $(\rho, c) = (10, 10)$. Both algorithms terminated when

$$\frac{\sum_{i=1}^N ||x_i^\nu - z^\nu||_2}{N||z^\nu||_2} \leq 10^{-6} \text{ and } \frac{|F(z^\nu) - F(z^*)|}{\max\{1, |F(z^*)|\}} \leq 10^{-10}$$

are satisfied. $F(z^*)$ is the optimal solution of (2.75) derived by running iADMM for 2000 iterations.

The computational results are presented in Table 2.1. The datasets are listed in the first column. The numbers of partitions N and the inexactness parameter γ are given in columns two and three separately. The ∞ symbol in the third column represents the exact x_i updates achieved by setting $\epsilon_\nu = 1e-10$ in all iterations. The average number of iterations (upon round off) for iADA and iADMM are given in the next two columns. The total amount of L-BFGS updates for both methods are presented in columns 6-7 and the average CPU time (in seconds) for these methods are given in the last two columns.

From Table 2.1, we see that when $\gamma = 1.5$ or 2.0, iADA shows better performance in the case $N = 20$ while iADMM converges faster when $N = 50$. For both algorithms, the CPU time is much longer in the case of $\gamma = 1.0$ when the convergence is not guaranteed in theory. Finally, compared with the exact update, it takes more iterations for the inexact version of both algorithms to converge but with shorter CPU time. This phenomenon results from the large number of L-BFGS updates

Table 2.1: Comparison of iADA and iADMM for solving (2.75).

Dataset	N	γ	Iteration		L-BFGS		CPU time	
			iADA	iADMM	iADA	iADMM	iADA	iADMM
w8a	20	1.0	274	380	70361	83703	**24.00**	30.00
		1.5	169	197	41089	58199	**14.18**	19.15
		2.0	164	195	44616	65647	**15.02**	20.87
		∞	150	133	49945	54928	**15.70**	17.78
	50	1.0	172	211	88460	127538	**16.00**	22.34
		1.5	140	120	78594	72673	13.10	**10.66**
		2.0	99	88	67909	60368	11.75	**10.10**
		∞	106	70	77627	62893	13.19	**10.50**
ijcnn1	20	1.0	202	276	49378	79120	**17.02**	26.80
		1.5	114	135	29741	42142	**10.16**	14.10
		2.0	112	134	31308	46742	**10.50**	15.11
		∞	190	186	73111	73193	23.02	**22.15**
	50	1.0	106	228	68001	115891	**11.74**	22.05
		1.5	107	112	58093	64195	**10.97**	11.58
		2.0	99	88	57777	50099	10.69	**9.27**
		∞	95	83	68652	69291	11.32	**11.21**

in each iteration of exact ADA and ADMM.

2.7 Conclusions

In this chapter, we study the convergence results of the ADA and its inexact version, the iADA, for solving multi-block separable convex minimization problems subject to linear constraints. First, we prove the global convergence and the $o(1/\nu)$ rate for the exact ADA when there exists a saddle point for the corresponding Lagrangian function. Next, global convergence and local linear convergence for the iADA are established under some mild assumptions and certain approximation criteria.

Before ending this chapter, we would like to discuss two possible directions related to the ADA. Firstly, we notice that both the primal PPA [34] and the Augmented Lagrangian Method [42] can be accelerated by utilizing the idea from Nesterov's seminal work [72]. It is natural to ask whether we can accelerate the ADA based on similar techniques since all of them belong to the general PPA framework. Secondly, the applicability of the approximation criteria in (A) and (B) is limited in practice due to the summable requirement and more implementable approximation criteria are needed for practical problems.

CHAPTER 3: Convergence of Multi-Block ADMM

3.1 Introduction

The alternating direction method of multipliers (ADMM) is widely used in statistics, machine learning and engineering while classical convergence results only work for convex and two-block optimization problems. The extension of ADMM to multi-block and nonconvex problems receive more attentions not only because of wide applications in statistics and machine learning but also its theoretical interests. In this chapter, we propose a multi-block two-level ADMM algorithm that solves nonconvex and nonsmooth linearly constrained optimization problem. We will discuss some intrinsic drawbacks of multi-block ADMM. Later, we will provide a two-level algorithm as a remedy to solve multi-block problems. Finally, some extensions will be further explored.

3.1.1 Problem

We consider the following abstraction problem

$$\min_{x_0,x_1,\ldots,x_K} \quad f(\mathbf{x}) = \sum_{i=0}^{K} f_i(x_i) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{i=0}^{K} A_i x_i = b, \tag{3.1b}$$

$$x_i \in \chi_i, i = 0, \ldots, K \tag{3.1c}$$

where $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{\infty\}$ is a continuous function, $x_i \in \chi_i \subset \mathbb{R}^{n_i}$ with the coefficient matrix $A_i \in \mathbb{R}^{m \times n_i}$. We set $b = 0$ throughout the paper to simplify the analysis and all of our results still hold if $b \neq 0$.

Our variable is $\mathbf{x} := [x_0; \ldots; x_K] \in \mathbb{R}^n$ where $n = \sum_{i=0}^{K} n_i$. Let $\mathbf{A} := [A_0 \ \cdots \ A_K] \in \mathbb{R}^{m \times n}$, $\mathbf{Ax} := \sum_{i=0}^{K} A_i x_i \in \mathbb{R}^m$, and $\chi := \Pi_{i=0}^{K} \chi_i$. To present the multi-block ADMM for the above

Algorithm 3.1 Multi-block ADMM for (3.1)

Initialize $x_1^0, \ldots, x_p^0, w^0$
while stopping criteria not satisfied **do**
  for $i = 0, \ldots, K$ **do**
    $x_i^{k+1} \leftarrow \text{argmin}_{x_i \in \chi_i} \mathcal{L}_\beta(x_{<i}^{k+1}, x_i, x_{>i}^k, w^k)$;
  $w^{k+1} \leftarrow w^k + \beta \mathbf{A}\mathbf{x}^{k+1}$;
  $k \leftarrow k + 1$;
return $x_0^k, \ldots, x_K^k$.

problem, we define the augmented Lagrangian:

$$\mathcal{L}_\beta(\mathbf{x}, w) := f(\mathbf{x}) + \langle w, \mathbf{A}\mathbf{x} \rangle + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}\|^2. \tag{3.2}$$

The Algorithm 3.1 extends the standard ADMM to multiple variable blocks. It also extends the *coordinate descent* algorithms dealing with linear constraints. We let $x_{<i} := [x_0; \ldots; x_{i-1}] \in \mathbb{R}^{n_0+n_1+\cdots+n_{i-1}}$ and $x_{>i} := [x_{i+1}; \ldots; x_K] \in \mathbb{R}^{n_{i+1}+\cdots+n_K}$ (clearly, $x_{<0}$ and $x_{>K}$ are null variables, which may be used for notational ease). Subvectors $x_{\leq i} := [x_{<i}, x_i]$ and $x_{\geq i}$ are defined similarly.

In spite of the success of ADMM on convex problems with two blocks (K=1), the behavior of multi-block ADMM on nonconvex problems has been largely a mystery, especially when there are also nonsmooth functions and nonconvex functions in the problems. It has been noticed that the direct extension of ADMM for the multi-block problem is not necessarily convergent even in the convex case [17]. The authors in [92] pointed out that two crucial conditions are needed for the global subsequential convergence of ADMM to a stationary point of (3.1)

- Condition 1: $\text{Im}[A_0, A_1, \ldots, A_{K-1}] \subseteq \text{Im}A_K$.
- Condition 2: The last block objective function $f_K(x_K)$ needs to be Lipschitz differentiable.

If any of the above conditions is violated, divergent examples of ADMM can be found in [47, 104]. We present two examples when these two conditions are not satisfied and ADMM diverges.

3.1.2 Two counter examples

Example 1: Consider the following convex problem in $\mathbb{R}^3$,

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & 0 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 0 \\ & x_1 + x_2 + 2x_3 = 0 \\ & x_1 + 2x_2 + 2x_3 = 0. \end{aligned} \tag{3.3}$$

Clearly, the only optimal solution is $(0, 0, 0)$ with optimal value 0. However, if we apply the three-block ADMM, the output will diverge for any $\beta > 0$. The following figure shows that the sequence generated by ADMM diverges to ∞ when $\beta = 1$. Even if we impose a compact constraint on the variables, the sequence still cannot converge to the optimal solution $(0, 0, 0)$. In this example, we can see that Condition 1 is violated as

$$\text{Im} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \nsubseteq \text{Im} \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

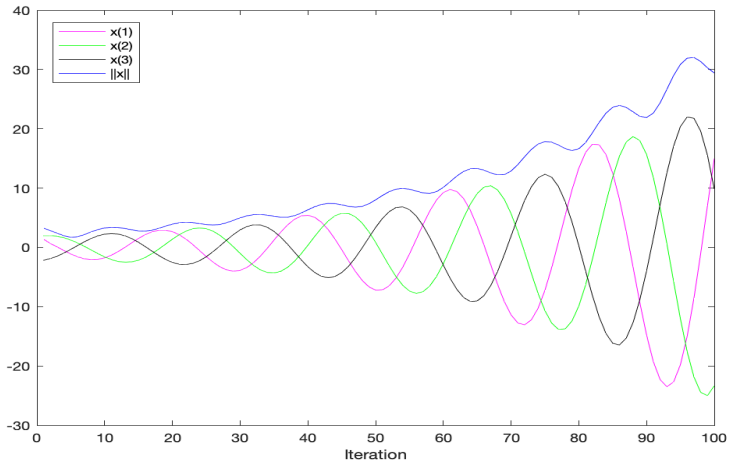


Figure 3.1: Counter example: condition 1 is violated.

Example 2: Consider the following nonconvex and nonsmooth problem in $\mathbb{R}^3$,

$$\begin{aligned} \min_{x\in\mathbb{R}^3} \quad & -|x_1|+|x_2|+|x_3| \\ \text{s.t.} \quad & x_1-x_2-x_3=0. \end{aligned} \tag{3.4}$$

The optimal value of the above problem is 0 and can be attained with infinite optimal solutions. However, for any penalty parameter $\beta > 0$, if we apply the three-block ADMM with $\beta > 0, (x_1^0, x_2^0, x_3^0, y^0) = (-2/\beta, 0, 0, -1)$, it will generate

$$(x_1^{2k}, x_2^{2k}, x_3^{2k}, y^{2k}) = (-2/\beta, 0, 0, -1)$$

and

$$(x_1^{2k+1}, x_2^{2k+1}, x_3^{2k+1}, y^{2k+1}) = (2/\beta, 0, 0, 1)$$

for sufficiently large k. The nonsmoothness of the final block $|x_3|$ which violates Condition 2 makes the algorithm unable to control the dual updates and therefore Algorithm 3.1 outputs jump between two points.

3.1.3 Our contributions

The authors in [92] firstly proposed the two-level idea to solve the two-block problem using ADMM. The two-level ADMM in this chapter is significantly different from their algorithm and we summarize our contributions as follows:

- We propose a novel two-level ADMM algorithm to solve the multi-block problem without assuming Condition 1 or 2. Notice that the theoretical development from two-block to multi-block is nontrivial.
- In our framework, the target function is no longer restricted in smooth functions while the theoretical results in [92] only apply to the smooth function. This indicates our algorithm can include important applications in statistics and machine learning, e.g., the ℓ_1-regularized problem.
- We evaluate the performance of the two-level ADMM by solving robust principal component

analysis and compress sensing. The two-level ADMM performs more robust than Algorithm 3.1 while not losing the convergence speed.

3.2 Prior work on multi-block ADMM

In the context of convex problems, the author in [47] studied problem (3.1) when each f_i is a (possibly nonsmooth) convex function. They prove that the multi-block ADMM will converge linearly to the global minimum assuming that a certain error bound condition holds true and the dual stepsize is sufficiently small. The numerical efficiency of Algorithm 3.1 has been demonstrated empirically in the literature, see e.g. [95, 75]. In [40, 41], the authors proposed some algorithms whose common feature is generating a new iterate by correcting the output of Algorithm 3.1 with some correction steps, instead of using the output of Algorithm 3.1. However, these algorithms are less numerically efficient than Algorithm 3.1 mainly because their correction steps need to determine step sizes iteratively with nonnegligible computation.

In the nonconvex world, one of the most general frameworks for proving convergence of ADMM on nonconvex problems is proposed by Wang et al. [104], where the following multi-block linearly constrained problem is studied, i.e.,

$$\min_{x_0,\cdots,x_K,z} \quad \sum_{i=0}^{K} f_i(x_i) + h(z) \quad \text{s.t.} \quad A_0x_0 + A_1x_1 + \cdots + A_Kx_K + Bz = b. \tag{3.5}$$

Variables $x_0, x_1, \ldots, x_p$ and z are the blocks and are updated in this order in the multi-block ADMM algorithm. Objective functions f_i's are continuous and possibly nonsmooth. Both f_i and h can be nonconvex. Not surprisingly, Condition 1 and 2 are crucial assumptions under this framework in order to show the convergence results.

The paper [48] studied nonconvex consensus and sharing problems. The concensus problem is given by

$$\min \quad h(x_0) + \sum_{i=1}^{K} g_k(x_k) \quad \text{s.t.} \quad x_i = x_0, i = 1, \ldots, K, \tag{3.6}$$

while the sharing problem is written as

$$\min \quad h(x_0) + \sum_{i=1}^{K} g_k(x_k) \quad \text{s.t.} \quad \sum_{k=1}^{K} A_k x_k = x_0. \tag{3.7}$$

Condition 1 is naturally inherited from the problem structure and the analysis relies on the assumption that $g_i(\cdot)$ is Lipschitz differentiable which is stronger than condition 2.

Other researches [56, 67] consider the variants of multi-block ADMM by linearizing the Lagrangian function in each block minimization step. While these variants may save the computational efforts in minimization steps, the theoretical analysis cannot avoid the Conditions 1 and 2.

3.3 The two-level ADMM: a remedy for the multi-block ADMM

3.3.1 A key reformulation and relaxation

We consider the following reformulation of (3.1):

$$\min_{\mathbf{x}\in\chi,\, z} \quad f(\mathbf{x}) \tag{3.8a}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + z = 0 \tag{3.8b}$$

$$z = 0. \tag{3.8c}$$

There is no doubt that problems (3.1) and (3.8) are equivalent to each other. The idea of adding a slack variable $z \in \mathbb{R}^m$ has two significant consequences. The first consequence is that the linear coupling constraint (3.8b) has $K + 2$ blocks, and the last block is an identity matrix $\mathbf{I}_m$, whose image is the whole space $\mathbb{R}^m$. Given any $\mathbf{x}$, there always exists z such that (3.8b) is satisfied. The second consequence is that constraint (3.8c) can be treated separately from (3.8b). If we ignore (3.8c) for a while, existing techniques in ADMM analysis can be applied to the rest of the problem. Since we want to utilize the unconstrained optimality condition of the last block, we can relax (3.8c). This observation motivates us to choose the classic powerful augmented Lagrangian method

(ALM). To be more specific, consider the problem

$$\min_{\mathbf{x}\in\chi,\, z} \quad f(\mathbf{x}) + (\mu^k)^\top z + \frac{\rho^k}{2}\|z\|^2 \tag{3.9a}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + z = 0 \tag{3.9b}$$

which is obtained by dualizing constraint (3.8c). The augmented Lagrangian term

$$g_k(z) = (\mu^k)^\top z + \frac{\rho^k}{2}\|z\|^2. \tag{3.10}$$

can be viewed as an objective function in variable z, which is not only Lipschitz differentiable but also strongly convex. Problem (3.9) can be solved by a $K+2$-block ADMM sequentially. Notice that the first order optimality condition of problem (3.9) at a stationary solution $(\mathbf{x}^k, z^k, w^k)$ is

$$0 \in \partial f(\mathbf{x}^k) + \mathbf{A}^\top w^k + \mathbf{N}_\chi(\mathbf{x}^k) \tag{3.11a}$$

$$\mu^k + \rho^k z^k + w^k = 0 \tag{3.11b}$$

$$\mathbf{A}\mathbf{x}^k + z^k = 0. \tag{3.11c}$$

However, such a solution may not necessarily satisfy constraint (3.8c), since we have intentionally *forgotten* it in the relaxation (3.9). Fortunately, the ALM offers a scheme to drive the slack variable z to zero by updating μ as follows:

$$\mu^{k+1} \leftarrow \mu^k + \rho^k z^k. \tag{3.12}$$

We can expect iterates to converge to a stationary point of the original problem (3.1). In summary, reformulation (3.8) separates the complication of the original problem into two levels, where the first level (3.9) provides a formulation that simultaneously satisfies Conditions 1 and 2, and the outer level ALM update will drive z to zero. Based on this idea, we propose a two-level ADMM to solve the original multi-block problem.

3.3.2 The two-level ADMM

The proposed algorithm consists of two levels, both of which are based on the augmented Lagrangian framework. The inner-level (indexed by t) uses multi-block ADMM to solve problem (3.9). Given $w \in \mathbb{R}^m$ and $\beta > 0$, the augmented Lagrangian function associated with the k-th inner-level problem (3.9) is defined as

$$\mathcal{L}_\beta^k(\mathbf{x}, z, w) := f(\mathbf{x}) + g_k(z) + \langle w, \mathbf{A}\mathbf{x} + z \rangle + \frac{\beta}{2} \|\mathbf{A}\mathbf{x} + z\|^2 . \tag{3.13}$$

See Algorithm 3.2 for the implementation fo the inner-level ADMM. Algorithm 3.2 will terminate if we find $(\mathbf{x}^t, z^t, w^t)$ such that

$$\sum_{i=1}^{K} \|A_i(x_i^{t-1} - x_i^t)\| \le \epsilon_1^k, \tag{3.14a}$$

$$\|z^{t-1} - z^t\| \le \epsilon_2^k. \tag{3.14b}$$

$$\|\mathbf{A}\mathbf{x}^t + z^t\| \le \epsilon_3^k. \tag{3.14c}$$

The above stopping criteria quantify the optimality and feasibility for an approximate stationary solution to the inner-level problem (3.9).

Algorithm 3.2 The k-th inner-level ADMM for (3.9)

Initialize $x_0^0, x_1^0, \ldots, x_K^0, z^0, w^0$ such that $\mu^k + \rho^k z^0 + w^0 = 0$, tolerance $\epsilon_i^k, i \in [3]$; index $t \leftarrow 1$;
while stopping criteria (3.14) is not satisfied **do**
  for $i = 0, \ldots, K$ **do**
    $x_i^{t+1} \leftarrow \operatorname{argmin}_{x_i \in \chi_i} \mathcal{L}_\beta^k(x_{<i}^{t+1}, x_i, x_{>i}^t, z^t, w^t)$;
  $z^{t+1} \leftarrow \operatorname{argmin}_z \mathcal{L}_\beta^k(\mathbf{x}^{t+1}, z, w^t)$;
  $w^{t+1} \leftarrow w^t + \beta\left(\mathbf{A}\mathbf{x}^{t+1} + z^{t+1}\right)$;
  $t \leftarrow t + 1$;
return $(\mathbf{x}^t, z^t, w^t)$.

We will prove in the next section that such criteria for Algorithm 3.2 is guaranteed to be met under certain conditions for the original problem (3.1). While the output of the inner-level ADMM will not satisfy the constraint (3.8c), the following outer-level ALM algorithm attempts to update the dual variables (μ^k, ρ^k) and finally push z^k to 0. See Algorithm 3.3 below.

Algorithm 3.3 Outer-level ALM

Initialize starting points $\mathbf{x}^0 \in \mathbb{R}^n$, $z^0 \in \mathbb{R}^m$;
  dual variable and bounds $\mu^0 \in [\underline{\mu}, \bar{\mu}]$ where $\underline{\mu}, \bar{\mu} \in \mathbb{R}^m$ and $\bar{\mu} - \underline{\mu} \in \mathbb{R}^m_{++}$;
  initial penalty parameter (μ^1, ρ^1), constants $\eta \in [0, 1)$ and $\tau > 1$;
  sequences of tolerance $\{\epsilon_i^k\} \subset \mathbb{R}_+$ with $\lim_{k\to\infty} \epsilon_i^k = 0$ for $i \in [3]$; index $k \leftarrow 1$;
while some stopping criterion is not satisfied **do**
  /* Inner level problem */
  given (μ^k, ρ^k), initialize Algorithm 3.2 with $(\mathbf{x}^{k-1}, z^{k-1})$ and denote the output by $(\mathbf{x}^k, z^k, w^k)$;
  /* Outer dual variable update */
  if $\|z^k\| \geq \eta\|z^{k-1}\|$ **then**
    $\mu^{k+1} \leftarrow \mu^k$, $\rho^{k+1} \leftarrow \tau\rho^k$;
  else
    $\mu^{k+1} \leftarrow \text{proj}_{[\underline{\mu},\bar{\mu}]}(\mu^k + \rho^k z^k)$, $\rho^{k+1} \leftarrow \rho^k$;
  $k \leftarrow k + 1$;

Remark: For Algorithm 3.3, to avoid the extremely large penalty parameter ρ^k in practice, one can increase the ρ^k only if $\|z^k\| \geq \max\{\eta\|z^{k-1}\|, \delta\}$ for some small tolerance δ of z^k.

3.4 Convergence results of the inner-level ADMM

In this section, we will show that by applying Algorithm 3.2, we are able to obtain an approximate stationary solution $(\mathbf{x}^k, z^k, w^k)$ satisfying

$$d_1^k \in \nabla f(\mathbf{x}^k) + \mathbf{A}^\top w^k + \mathbf{N}_\chi(\mathbf{x}^k) \tag{3.15a}$$

$$\mu^k + \rho^k z^k + w^k = 0 \tag{3.15b}$$

$$\mathbf{A}\mathbf{x}^k + z^k = d_2^k. \tag{3.15c}$$

such that $\left\|d_i^k\right\| \to 0, i = 1, 2$. We consider the k-th inner-level problem, where the outer-level dual variables (μ^k, ρ^k) and the augmented Lagrangian function $\mathcal{L}_\beta^k$ are abbreviated as (μ, ρ) and $\mathcal{L}_\beta$ respectively, if not specified explicitly. To save space, throughout this section we let

$$(\mathbf{x}^+, z^+, w^+) := (\mathbf{x}^{t+1}, z^{t+1}, w^{t+1}). \tag{3.16}$$

To ensure the convergence of the sequence $(\mathbf{x}^t, z^t, w^t)$, we need to following assumption.

Assumption 3.1. *The feasible sets $\chi_i, i = 0, \ldots, K$, are compact convex sets.*

Assumption 3.2. *For $i = 0, \ldots, K, A_s$ has full column rank so that $\sigma_i := \sigma_{\min}(A_i^T A_i) > 0$ where $\sigma_{\min}$ denotes the minimum eigenvalue of a matrix..*

Assumption 3.3 (objective f regularity)**.** *(a) f_0 is continuous and the supremum* $\sup\{\|d\| : x_0 \in \chi_0, d \in \partial f_0(x_0)\}$ *is bounded.*

(b) For $i = 1, \ldots, K$, f_i is either closed proper convex (possibly nonsmooth) or γ_i-smooth and define

$$\tilde{\gamma}_i = \begin{cases} \gamma_i, & \text{if } f_i \text{ is } \gamma_i - \text{smooth} \\ 0, & \text{if } f_i \text{ is convex.} \end{cases}$$

We have the following remarks regarding to the assumptions made.

- Assumption 3.1 requires the feasible set of the variables to be compact. This condition is not needed in conventional analysis of ADMM, but is required here to ensure the boundedness of the outputs. This assumption is usually satisfied in practical applications (e.g. the consensus problems) whenever a priori knowledge on the variable domain is available.

- Assumption 3.2 is standard and made to ensure the subproblems has a unique minimizer. It can be relaxed if we can guarantee the Lipschitz sub-minimization paths defined in [104].

- Assumption 3.3 discusses the regularity of the objective function. We notice that the first block f_0 is more flexible compared with the following K blocks. Our framework include broad practical problems including penalized regression, classification, compressed sensing and more as the functions f_i's can be nonconvex and nonsmooth.

Lemma 3.1 (bound dual by primal)**.** *For all $t \in \mathbb{N}$, it holds that*

(a) $\mu + \rho z^t + w^t = 0$.

(b) $\|w^+ - w^t\| = \rho\|z^+ - z^t\|$.

Proof. Notice that $\nabla g(z) = \mu + \rho z$. Part (a) follows directly from the optimality condition of z^t: $0 = \nabla g(z^t) + w^{t-1} + \beta(A\mathbf{x}^t + z^t)$, and $w^t = w^{t-1} + \beta\left(A\mathbf{x}^t + z^t\right)$. For Part (b). Since

$w^+ - w^t = \beta(A\mathbf{x}^+ + z^+)$, we get

$$\|w^+ - w^t\| = \|\nabla g(z^+) - \nabla g(z^t)\| = \rho\|z^+ - z^t\|.$$

□

The next lemma provides estimations for the descent of the augmented Lagrangian function $\mathcal{L}_\beta(\mathbf{x}, z, w)$. Because of the optimality of x_i^t, we can introduce the following subgradient d_i^t,

$$d_i^t := -(A_i^T z^+ + \beta\rho_i^t) \in \partial_i f_i(x_i^+) + \mathbf{N}_{\chi_i}(x_i) \tag{3.17}$$

where

$$\rho_i^t := A_i^T(A_{>i}x_{>i}^t - A_{>i}x_{>i}^+) + A_i^T(z^t - z^+).$$

Lemma 3.2 (descent of $\mathcal{L}_\beta$ during x_i update)**.** *The iterates in Algorithm 3.2 satisfy*

1. $\mathcal{L}_\beta(x_{<i}^+, \boldsymbol{x}_i^t, x_{>i}^t, z^t, w^t) \geq \mathcal{L}_\beta(x_{<i}^+, \boldsymbol{x}_i^+, x_{>i}^t, z^t, w^t)$, $i = 0, \ldots, K$;

2. $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) \geq \mathcal{L}_\beta(\mathbf{x}^+, z^t, w^t)$;

3. $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^+, z^t, w^t) = \sum_{i=0}^K r_i$, *where*

$$r_i := f_i(x_i^t) - f_i(x_i^+) - \left\langle d_i^t, x_i^t - x_i^+ \right\rangle + \frac{\beta}{2}\|A_i x_i^t - A_i x_i^+\|^2 \geq 0, \tag{3.18}$$

where d_i^t is defined in (3.17)*. For $i = 1, \ldots, K$, under Assumptions 3.2-3.3,*

$$r_i \geq \frac{\beta - \tilde{\gamma}_i/\sigma_i}{2}\|A_i x_i^t - A_i x_i^+\|^2; \tag{3.19}$$

Proof. **Part 1** follows directly from the minimization subproblems, which give x_i^+.

Part 2 is a result of

$$\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^+, z^t, w^t) = \sum_{i=0}^K \big(\mathcal{L}_\beta(x_{<i}^+, x_i^t, x_{>i}^t, z^t, w^t) - \mathcal{L}_\beta(x_{<i}^+, x_i^+, x_{>i}^t, z^t, w^t)\big), \tag{3.20}$$

and part 1.

Part 3: Each term in the sum equals $f(x_i^t) - f(x_i^+)$ *plus*

$$\begin{aligned}
&\left\langle w^t, A_i x_i^t - A_i x_i^+ \right\rangle + \frac{\beta}{2}\|A_{<i}x_{<i}^+ + A_i x_i^t + A_{>i}x_{>i}^t + z^t\|^2 - \frac{\beta}{2}\|A_{<i}x_{<i}^+ + A_i x_i^+ + A_{>i}x_{>i}^t + z^t\|^2 \\
&= \left\langle w^t, A_i x_i^t - A_i x_i^+ \right\rangle + \left\langle \beta\left(A_{<i}x_{<i}^+ + A_i x_i^+ + A_{>i}x_{>i}^t + z^t\right), A_i x_i^t - A_i x_i^+ \right\rangle + \frac{\beta}{2}\|A_i x_i^t - A_i x_i^+\|^2 \\
&= \left\langle A_i^T w^+ + \beta\rho_i^t, x_i^t - x_i^+ \right\rangle + \frac{\beta}{2}\|A_i x_i^t - A_i x_i^+\|^2
\end{aligned} \tag{3.21}$$

where the first equality follows from the cosine rule: $\|b+c\|^2 - \|a+c\|^2 = \|b-a\|^2 + 2\langle a+c, b-a\rangle$ with $b = A_i x_i^t$, $a = A_i x_i^+$, and $c = A_{<i}x_{<i}^+ + A_{>i}x_{>i}^t + z^t$. Let d_i^t be defined in (3.17). From Assumptions 3.3, if f_i is γ_i-smooth, we get

$$f_i(x_i^t) - f_i(x_i^+) - \left\langle d_i^t, x_i^t - x_i^+ \right\rangle \geq -\frac{\gamma_i}{2}\|x_i^t - x_i^+\|^2 \geq -\frac{\gamma_i/\sigma_i}{2}\|Ax_i^t - Ax_i^+\|^2. \tag{3.22}$$

If f_i is convex, then $r_i \geq \frac{\beta}{2}\|A_i x_i^t - A_i x_i^+\|^2$ holds trivially. □

Lemma 3.3. *(descent of $\mathcal{L}_\beta$ due to z and w updates) For any $t \in \mathbb{N}$*

$$\mathcal{L}_\beta(\mathbf{x}^+, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) \geq \left(\frac{\rho+\beta}{2} - \frac{\rho^2}{\beta}\right)\|z^+ - z^t\|^2. \tag{3.23}$$

Proof. It follows from Lemma 3.1 that

$$\begin{aligned}
&\mathcal{L}_\beta(\mathbf{x}^+, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) \\
&= g(z^t) - g(z^+) + \left\langle w^+, z^t - z^+ \right\rangle + \frac{\beta}{2}\|z^+ - z^t\|^2 - \beta\|\mathbf{A}\mathbf{x}^+ + z^+\|^2 \\
&= \frac{\rho}{2}\|z^+ - z^t\|^2 + \frac{\beta}{2}\|z^+ - z^t\|^2 - \frac{1}{\beta}\|w^+ - w^t\|^2 \\
&= \left(\frac{\rho+\beta}{2} - \frac{\rho^2}{\beta}\right)\|z^+ - z^t\|^2.
\end{aligned}$$

The first equality is due to $-\beta(a+b)^\top(a+c) + \frac{\beta}{2}\|a+c\|^2 - \frac{\beta}{2}\|a+b\| = \frac{\beta}{2}\|c-b\|^2 - \beta\|a+b\|^2$ with $a = \mathbf{A}\mathbf{x}^+, b = z^+$, and $c = z^t$; the second inequality is a result of the fact that $g(z)$ is a quadratic function. □

Lemma 3.4 (Monotone, lower–bounded $\mathcal{L}_\beta$ and bounded sequence)**.** *Suppose that $\beta > \rho$, then the sequence $(\mathbf{x}^t, z^t, w^t)$ generated by Algorithm 3.2 satisfies*

1. $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) \geq \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+)$.
2. $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t)$ *is lower bounded for all $t \in \mathbb{N}$ and converges as $t \to \infty$.*
3. $\{\mathbf{x}^t, z^t, w^t\}$ *is bounded.*

Proof. **Part 1**. It is a direct result of Lemma 3.2 part 2, and Lemma 3.3.

Part 2. Let $z' = -\mathbf{A}\mathbf{x}^t$, by Assumption 3.1 and $\rho > 0$, we have

$$f(\mathbf{x}^t) + g(z') > -\infty.$$

Then we have

$$\begin{aligned}
\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) &= f(\mathbf{x}^t) + g(z^t) + \langle w^t, z^t - z' \rangle + \frac{\beta}{2} \left\| \mathbf{A}\mathbf{x}^t + z^t \right\|^2 \\
&= f(\mathbf{x}^t) + g(z^t) + \langle \nabla g(z^t), z' - z^t \rangle + \frac{\beta}{2} \left\| \mathbf{A}\mathbf{x}^t + z^t \right\|^2 \\
(g \text{ is quadratic}) \quad &= f(\mathbf{x}^t) + g(z') - \frac{\rho}{2} \left\| z' - z^t \right\|^2 + \frac{\beta}{2} \left\| \mathbf{A}\mathbf{x}^t + z^t \right\|^2 \\
(\beta > \rho) \quad &= f(\mathbf{x}^t) + g(z') + \frac{\beta - \rho}{2} \left\| \mathbf{A}\mathbf{x}^t + z^t \right\|^2 > -\infty.
\end{aligned}$$

Part 3. From parts 1 and 2, $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t)$ is upper bounded by $\mathcal{L}_\beta(\mathbf{x}^0, z^0, w^0)$ and so are $f(\mathbf{x}^t) + g(z')$ and $\|\mathbf{A}\mathbf{x}^t + z^t\|^2$. By Assumption 3.1, $\{\mathbf{x}^t\}$ is bounded and, therefore $\{z^t\}$ is also bounded. By Lemma 3.1, $\{w^t\}$ is also bounded. □

Lemma 3.5 (Asymptotic regularity)**.** $\lim_{k\to\infty} \|z^t - z^+\| = 0$ *and* $\lim_{t\to\infty} \|w^t - w^+\| = 0$.

Proof. The first result follows directly from Lemmas 3.2, 3.3, and 3.4 part (2), and the second limit results from Lemma 3.1 part (b). □

Lemma 3.6 (Sufficient descent property)**.** *Suppose that*

$$\beta > \max\{\rho, \tilde{\gamma}_1/\sigma_1, \ldots, \tilde{\gamma}_K/\sigma_K\} + 2.$$

Then, Algorithm 3.2 satisfies the sufficient descent property

$$\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) \geq \sum_{i=1}^{K} \|A_i x_i^t - A_i x_i^+\|^2 + \|z^+ - z^t\|^2. \tag{3.24}$$

Proof. From Lemmas 3.2 and 3.3, we can obtain

$$\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) \geq \sum_{i=1}^{K} \frac{\beta - \tilde{\gamma}_i/\sigma_i}{2} \|A_i x_i^t - A_i x_i^+\|^2 + (\frac{\rho+\beta}{2} - \frac{\rho^2}{\beta}) \|z^+ - z^t\|^2. \tag{3.25}$$

If $\beta > \max\{\rho, \tilde{\gamma}_1/\sigma_1, \ldots, \tilde{\gamma}_K/\sigma_K\} + 2$, then

$$\frac{\rho+\beta}{2} - \frac{\rho^2}{\beta} > \rho + 1 - \frac{\rho^2}{\rho+1} = \frac{2\rho+1}{\rho+1} > 1.$$

(3.24) follows directly from above. □

Lemma 3.7 (subgradient bound property)**.** *There exists $C(\beta, \rho) > 0$ and $d \in \partial\mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+)$ such that*

$$\|d\| \leq C(\beta, \rho)\Big(\|z^+ - z^t\| + \sum_{i=1}^{K} \|A_i(x_i^+ - x_i^t)\|\Big). \tag{3.26}$$

Proof. Recall that $f(\mathbf{x}) = \sum_{i=0}^{K} f_i(x_i)$, we know

$$\partial\mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) = \left(\left\{\frac{\partial\mathcal{L}_\beta}{\partial x_i}\right\}_{i=0}^{K}, \nabla_z\mathcal{L}_\beta, \nabla_w\mathcal{L}_\beta\right)(\mathbf{x}^+, z^+, w^+).$$

In order to prove the lemma, we only need to show that each block of $\partial\mathcal{L}_\beta$ can be controlled by some constant depending on β. Therefore, it suffices to prove for $s = 0, \ldots, K$, there exists $d_s \in \frac{\partial\mathcal{L}_\beta}{\partial x_s}(\mathbf{x}^+, z^+, w^+)$ such that

$$\|d_s\| \leq (\sigma_{\max}(A_s)(\beta+\rho) + \frac{\rho}{\beta})\left(\sum_{i=1}^{p} \|A_i x_i^+ - A_i x_i^t\| + \|z^+ - z^t\|\right), \tag{3.27}$$

and

$$\|\nabla_w \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+)\| \leq \frac{\rho}{\beta}\|z^+ - z^t\|, \tag{3.28}$$

$$\|\nabla_z \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+)\| \leq \rho\|z^+ - z^t\|. \tag{3.29}$$

In order to prove (3.28), we notice that

$$\nabla_w \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) = \mathbf{A}\mathbf{x}^+ + z^+ = \frac{1}{\beta}(w^+ - w^t).$$

By Lemma 3.1,

$$\|\nabla_w \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+)\| = \frac{\rho}{\beta}\|z^+ - z^t\|.$$

In order to prove (3.29), we have $\nabla_z \mathcal{L}_\beta(\mathbf{x}^+, z^+, w^+) = w^+ - w^t$ and apply Lemma 3.1. In order to prove (3.27), observe that for $s = 0, \ldots, K$,

$$\begin{aligned}
&\frac{\partial \mathcal{L}_\beta}{\partial x_s}(\mathbf{x}^+, z^+, w^+) \\
=&\partial f_s(x_s^+) + \mathbf{N}_{\chi_s}(x_s^+) + A_s^T w^+ + \beta A_s^T \left(\mathbf{A}\mathbf{x}^+ + z^+\right) \\
=&\partial f_s(x_s^+) + \mathbf{N}_{\chi_s}(x_s^+) + A_s^T w^t + \beta A_s^T \left(A_{\leq s} x_{\leq s}^+ + A_{>s} x_{>s}^t + z^t\right) \\
&+ A_s^T(w^+ - w^t) + \beta A_s^T \left(A_{>s} x_{>s}^+ - A_{>s} x_{>s}^t + z^+ - z^t\right).
\end{aligned} \tag{3.30}$$

By the optimality condition for x_s^+,

$$0 \in \partial f_s(x_s^+) + \mathbf{N}_{\chi_s}(x_s^+) + A_s^T w^t + \beta A_s^T \left(A_{\leq s} x_{\leq s}^+ + A_{>s} x_{>s}^t + z^t\right).$$

Thus for $s = 0, \ldots, K$, we can define d_s as

$$\begin{aligned}
d_s &:= A_s^T(w^+ - w^t) + \beta A_s^T \left(A_{>s} x_{>s}^+ - A_{>s} x_{>s}^t + z^+ - z^t\right) \\
&\in \frac{\partial \mathcal{L}_\beta}{\partial x_s}(\mathbf{x}^+, z^+, w^+).
\end{aligned} \tag{3.31}$$

We notice that d_s does not involve with x_0^t for any s. $w^+ - w^t, A_{>s} x_{>s}^+ - A_{>s} x_{>s}^t$ and $z^+ - z^t$ can

be bounded by $\left(\sum_{i=1}^{p}\|A_i x_i^+ - A_i x_i^t\| + \|z^+ - z^t\|\right)$. Let $\sigma_{\max}(A_s)$ be the largest singular value of A_s, we can bound d_s by

$$\|d_s\| \leq (\sigma_{\max}(A_s)(\beta+\rho) + \frac{\rho}{\beta})\left(\sum_{i=1}^{K}\|A_i x_i^+ - A_i x_i^t\| + \|z^+ - z^t\|\right).$$

This completes the proof. □

We have established the following properties regarding the updates of Algorithm 3.2.

P1 **(Boundedness)** $\{\mathbf{x}^t, z^t, w^t\}$ is bounded, and $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t)$ is lower bounded.

P2 **(Sufficient descent)** There is a constant $C_1(\beta,\rho) > 0$ such that for all sufficiently large t, we have

$$\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t) - \mathcal{L}_\beta(\mathbf{x}^{t+1}, z^{t+1}, w^{t+1}) \geq C_1(\beta,\rho)\Big(\|z^{t+1} - z^t\|^2 + \sum_{i=1}^{K}\|A_i(x_i^t - x_i^{t+1})\|^2\Big). \tag{3.32}$$

P3 **(Subgradient bound)** There exists $C_2(\beta,\rho) > 0$ and $d^{t+1} \in \partial\mathcal{L}_\beta(\mathbf{x}^{t+1}, z^{t+1}, w^{t+1})$ such that

$$\|d^{t+1}\| \leq C_2(\beta,\rho)\Big(\|z^{t+1} - z^t\| + \sum_{i=1}^{K}\|A_i(x_i^{t+1} - x_i^t)\|\Big). \tag{3.33}$$

It is our intention to start i at 1, thus skipping the x_0-block, in (3.32) and (3.33).

Theorem 3.1. *Suppose that Assumptions 3.1-3.3 hold. Then, if*

$$\beta > \max\{\rho, \tilde{\gamma}_1/\sigma_1, \ldots, \tilde{\gamma}_K/\sigma_K\} + 2,$$

Algorithm 3.2 converges subsequently. And each limit point $(\mathbf{x}^*, z^*, w^*)$ *is a stationary point of* (3.11)*. Furthermore, the running best rates of the sequences* $\{\|z^{t+1} - z^t\|^2 + \sum_{i=1}^{K}\|A_i(x_i^t - x_i^{t+1})\|^2\}$ *and* $\{\|d^{t+1}\|\}$ *are* $o(\frac{1}{t})$ *and* $o(\frac{1}{\sqrt{t}})$*, respectively.*

Proof. Let $(\mathbf{x}^*, z^*, w^*)$ be the limit point of a sub-sequence $(\mathbf{x}^{t_s}, z^{t_s}, w^{t_s})$ for $s \in \mathbb{N}$. By Assumption 3.3, $\mathcal{L}_\beta(\mathbf{x}^*, z^*, w^*) = \lim_{s\to\infty}\mathcal{L}_\beta(\mathbf{x}^{t_s}, z^{t_s}, w^{t_s})$.

By **P1**, the sequence $(\mathbf{x}^t, z^t, w^t)$ is bounded and therefore there exists a convergent subsequence $\{t_s\}_{s=1}^{\infty}$,

$$(\mathbf{x}^{t_s}, z^{t_s}, w^{t_s}) \to (\mathbf{x}^*, z^*, w^*)$$

as $s \to \infty$. By **P1** and **P2**, $\mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t)$ is monotonically nonincreasing and lower bounded, and so

$$\|z^{t+1} - z^t\| \to 0, \quad \sum_{i=1}^{K} \|A_i(x_i^{t+1} - x_i^t)\| \to 0$$

as $t \to \infty$. From **P3**, there exists $d^t \in \mathcal{L}_\beta(\mathbf{x}^t, z^t, w^t)$ such that $\left\|d^t\right\| \to 0$. Then we have $0 \in \partial\mathcal{L}_\beta(\mathbf{x}^*, z^*, w^*)$.

The running best rate of the sequence $\{\|z^{t+1} - z^t\|^2 + \sum_{i=1}^{K} \|A_i(x_i^t - x_i^{t+1})\|^2\}$ can be obtained via [55, Lemma 3]. By **P3** the running best rate for $\{\left\|d^{t+1}\right\|\}$ is $o(\frac{1}{\sqrt{t}})$. □

By Theorem 3.1, The above theorem indicates our stopping criteria in (3.14) is guaranteed to be satisfied by Algorithm 3.2 and the best running rate is $o(\frac{1}{\sqrt{t}})$.

3.5 Convergence results of the outer-level ALM

In this section, we prove the convergence results of the outer-level updates.

Theorem 3.2. *Suppose that Assumptions 3.1-3.3 hold. Let $(\mathbf{x}^k, z^k, w^k)$ be the sequence of inner-level iterates satisfying condition* (3.15)*. Then the iterates of primal solutions $\{(\mathbf{x}^k, z^k)\}$ are bounded and every cluster point $(\mathbf{x}^*, z^*)$ of $\{(\mathbf{x}^k, z^k)\}$ satisfies either one of the following:*

(1) $\mathbf{x}^*$ is feasible for problem (3.1)*, i.e., $z^* = 0$.*

(2) $\mathbf{x}^*$ is a stationary point of the following problem:

$$\min_{\mathbf{x}\in\chi} \frac{1}{2} \|\mathbf{A}\mathbf{x}\|^2 . \tag{3.34}$$

Proof. Since $\mathbf{x}^k \in \chi$ and χ is bounded. From (3.15) and $\left\|\mathbf{A}\mathbf{x}^k + z^k\right\| \le \epsilon_2^k \to 0$, $\{z^k\}$ is also bounded. Therefore we may assume that the sequence $\{(\mathbf{x}^k, z^k)\}$ converges to $(\mathbf{x}^*, z^*)$. As χ is a compact set, the point $\mathbf{x}^* \in \chi$ and $\mathbf{A}\mathbf{x}^* + z^* = \lim_{k\to\infty} \mathbf{A}\mathbf{x}^k + z^k = 0$. If ρ^k is bounded, we have

$z^k \to 0$ based on the update rule in the outer-level ALM. It follows that $\mathbf{A}\mathbf{x}^* = 0$ which implies that $\mathbf{x}^*$ is feasible. Otherwise, if that ρ^k is unbounded, $\lim_{k\to\infty} \rho^k = \infty$. By dividing ρ^k on (3.15b), we have

$$\frac{\mu^k}{\rho^k} + z^k + \frac{w^k}{\rho^k} = 0.$$

From the update of μ^k in the outer level, $\{\mu^k\}$ is bounded and therefore we may assume $\mu^k \to \mu^*$. In one case, if $\{w^k\}$ is bounded, we have $z^* = 0$ as $k \to \infty$. Otherwise, $\lim_{k\to\infty} w^k = \infty$ and the sequence $\{\frac{w^k}{\rho^k}\}$ converges to some point $\hat{w}^*$ where

$$\hat{w}^* + z^* = 0.$$

By (3.15a), we obtain

$$d_1^k \in \partial f(\mathbf{x}^k) + \mathbf{A}^\top w^k + \mathbf{N}_\chi(\mathbf{x}^k).$$

Since the normal cone $\mathbf{N}_\chi(\mathbf{x}^k)$ is closed, by dividing ρ^k on both sides, we have

$$\frac{d_1^k}{\rho^k} - \mathbf{A}^\top(\frac{w^k}{\rho^k}) \in \frac{\partial f(\mathbf{x}^k)}{\rho^k} + \mathbf{N}_\chi(\mathbf{x}^k). \tag{3.35}$$

It is easy to see that $\partial f(\mathbf{x}^k) = \Pi_{i=0}^K \partial f_i(x_i^k)$. From Assumption 3.3(a), we know there exists a constant $M_0 > 0$ such that

$$\partial f_0(x_0^k) \subset [-M_0, M_0]^{n_0}, \qquad \forall k > 0.$$

For $i = 1, \ldots, K$, if f_i is γ_i-smooth, $\partial f_i(x_i^k)$ coincides with $\nabla f_i(x_i^k)$. Since χ_i is bounded, there exists a constant $M_i > 0$ such that $\left\|\nabla f_i(x_i^k)\right\| \leq M_i$. Otherwise, if f_i is a general convex function, from [79, Theorem 24.5], given any $\epsilon > 0$, there exists an index k_i such that

$$\partial f_i(x_i^k) \subset \partial f_i(x_i^*) + \epsilon\mathbf{B}, \qquad \forall k > k_i,$$

as $x_i^k \to x_i^*$. Similarly, we can find a positive constant $M_i > 0$ such that

$$\partial f_i(x_i^k) \subset [-M_i, M_i]^{n_i}, \qquad \forall k > k_i.$$

The above observations together prove that

$$\lim_{k\to\infty} \frac{\partial f(\mathbf{x}^k)}{\rho^k} \to \{\mathbf{0}\}$$

under Assumption 3.3. Taking the limit of (3.35) , we have

$$0 \in \mathbf{N}_\chi(\mathbf{x}^*) + \mathbf{A}^\top \hat{w}^* \tag{3.36a}$$

$$\hat{w}^* + z^* = 0 \tag{3.36b}$$

$$\mathbf{A}\mathbf{x}^* + z^* = 0 \tag{3.36c}$$

which implies that $\mathbf{x}^*$ is a stationary point of the quadratic optimization problem (3.34). □

Theorem 3.3. *Suppose that Assumptions 3.1-3.3 hold. Let (x^*, z^*) be a limit point of outer-level iterates $\{(x^k, z^k)\}$. If $\{w^k\}$ has a limit point w^* along the subsequence converging to (x^*, z^*). Then (x^*, w^*) is a stationary point of problem* (3.1) *satisfying the stationary condition*

$$0 \in \partial f(\mathbf{x}^*) + \mathbf{A}^\top w^* + \mathbf{N}_\chi(\mathbf{x}^*) \tag{3.37a}$$

$$\mathbf{A}\mathbf{x}^* = 0. \tag{3.37b}$$

Proof. Without the loss of generosity, we assume that the whole sequence $\{(x^k, z^k, w^k)\}$ converges to a limit point $(\mathbf{x}^*, z^*, w^*)$. Applying a similar argument in the proof of Theorem 3.2, we obtain that the limit $\mathbf{x}^* \in \chi$ and $\mathbf{A}\mathbf{x}^* + z^* = 0$. In order to prove the feasibility (3.37b), we need to show that $z^* = 0$. If ρ^k is bounded, we have $z^k \to 0$ and thus $z^* = 0$. Otherwise, ρ^k is unbounded. By taking the limit on both sides of the following equation

$$\frac{\mu^k}{\rho^k} + z^k + \frac{w^k}{\rho^k} = 0,$$

we can also derive the fact that $z^* = 0$, since μ^k and w^k are bounded. As $d_1^k \to 0$, by taking the limit on both sides of (3.15a), we can get (3.37a). □

3.6 Numerical experiments

3.6.1 Counter examples revisited

Now, let's use the proposed two-level ADMM algorithm to solve the two counter examples and verify the theoretical convergence results provided in the previous sections.

Example 1: For problem (3.3), we initialize the two-level ADMM with parameter $(\rho^0, \beta) = (1, 3)$ as suggested by Lemma 3.6. The following figure shows the output of the outer-level ALM. Clearly, we can see the algorithm converges to the optimal solution $(x_1, x_2, x_3) = (0, 0, 0)$ after a few iterations.

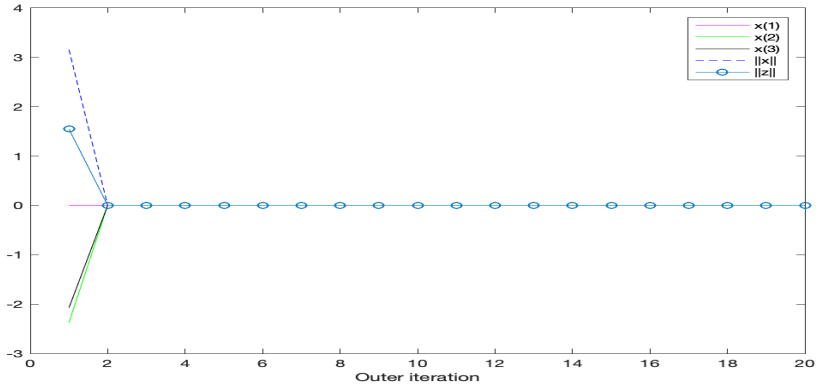


Figure 3.2: Counter example 1 converges with the two-level ADMM.

Example 2: For problem (3.4), we notice that the objective function $-|x_1| + |x_2| + |x_3|$ is nonsmooth but still under the framework of Assumption 3.3. We discover that the two-level ADMM will converge to the optimal value 0 while different initializations lead to different optimal solutions. This result is in line with our theoretical analysis in the previous section as there exist infinitely many solutions to the original problem.

3.6.2 Robust principle component analysis (RPCA)

RPCA is used to obtain a low rank and sparse decomposition of a given matrix $\mathbf{A}$ corrupted by noise [14]:

$$\min \frac{1}{2} \|\mathbf{X}_1\|_F^2 + \gamma_2 \|\mathbf{X}_2\|_1 + \gamma_3 \|\mathbf{X}_3\|_* \quad \text{s.t.} \quad \mathbf{A} = \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3. \tag{3.38}$$

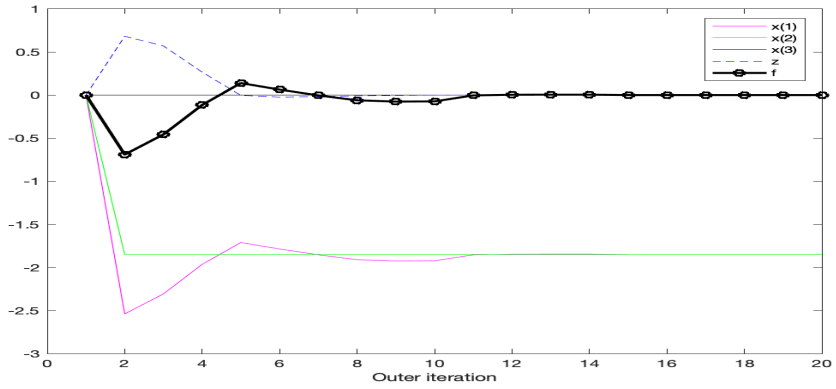


Figure 3.3: Counter example 2 converges with the two-level ADMM.

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{X}_1$ is a noise matrix, $\mathbf{X}_2$ is a sparse matrix and $\mathbf{X}_3$ is a low rank matrix. $\mathbf{A} = \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3$ is generated in the same way as [74]. Given the multiplier $\mathbf{B} \in \mathbb{R}^{m\times n}$ and penalty β, the augmented Lagrangian function for problem (3.38) is given by

$$\begin{aligned}\mathcal{L}_\beta(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3; \mathbf{B}) =& \frac{1}{2}\|\mathbf{X}_1\|_F^2 + \gamma_2\|\mathbf{X}_2\|_1 + \gamma_3\|\mathbf{X}_3\|_* \\ &+ \langle \mathbf{B}, \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 - \mathbf{A}\rangle + \frac{\beta}{2}\|\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 - \mathbf{A}\|_F^2 .\end{aligned} \tag{3.39}$$

Here, we provide the three-block ADMM updates and the two-level ADMM works similarly. The following two lemmas [74] are essential in the sequential updates of block minimization.

Lemma 3.8. *For $\mu > 0$ and $\mathbf{Y} \in \mathbb{R}^{m\times n}$, the solution of the following problem*

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \{\mu\|\mathbf{X}\|_1 + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2\},$$

is given by $\mathcal{S}_\mu(\mathbf{Y})$, which is defined componentwisely by

$$(\mathcal{S}_\mu(\mathbf{Y}))_{ij} := \max\{|\mathbf{Y}_{ij}| - \mu, 0\} \cdot \operatorname{sign}(\mathbf{Y}_{ij}). \tag{3.40}$$

Lemma 3.9. *The solution of the following problem*

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \{\mu\|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2\},$$

is given by $\mathcal{D}_\mu(\mathbf{Y})$*, which is defined by*

$$\mathcal{D}_\mu(\mathbf{Y}) := \mathbf{U}\text{diag}\left(\mathcal{S}_\mu(\Sigma)\right)\mathbf{V}^T, \tag{3.41}$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$*,* $\mathbf{V} \in \mathbb{R}^{n\times r}$ *and* $\Sigma \in \mathbb{R}^{r\times r}$ *are obtained by the singular value decomposition (SVD) of* $\mathbf{Y}$*:*

$$\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T, \quad \textit{and} \quad \Sigma = \text{diag}\left(\sigma_1, \ldots, \sigma_r\right).$$

Then, in each iteration $k+1$, the three-block ADMM updates as follows

$$\begin{aligned}
\mathbf{X}_1^{k+1} &= \underset{\mathbf{X}_1}{\text{argmin}}\, \mathcal{L}_\beta(\mathbf{X}_1, \mathbf{X}_2^k, \mathbf{X}_3^k; \mathbf{B}^k) = \frac{1}{1+\beta}(\beta(\mathbf{A} - \mathbf{X}_2^k - \mathbf{X}_3^k) - \mathbf{B}^k),\\
\mathbf{X}_2^{k+1} &= \underset{\mathbf{X}_2}{\text{argmin}}\, \mathcal{L}_\beta(\mathbf{X}_1^{k+1}, \mathbf{X}_2, \mathbf{X}_3^k; \mathbf{B}^k) = \mathcal{S}_{\gamma_2/\beta}(\mathbf{A} - \mathbf{X}_1^{k+1} - \mathbf{X}_3^k - \frac{\mathbf{B}^k}{\beta}),\\
\mathbf{X}_3^{k+1} &= \underset{\mathbf{X}_3}{\text{argmin}}\, \mathcal{L}_\beta(\mathbf{X}_1^{k+1}, \mathbf{X}_2^{k+1}, \mathbf{X}_3; \mathbf{B}^k) = \mathcal{D}_{\gamma_3/\beta}(\mathbf{A} - \mathbf{X}_1^{k+1} - \mathbf{X}_2^{k+1} - \frac{\mathbf{B}^k}{\beta}),\\
\mathbf{B}^{k+1} &= \mathbf{B}^k + \beta(\mathbf{X}_1^{k+1} + \mathbf{X}_2^{k+1} + \mathbf{X}_3^{k+1} - \mathbf{A}).
\end{aligned} \tag{3.42}$$

Figure 3.4 compares the convergence results of the two-level ADMM with multi-block ADMM in terms of the objective value and feasibility. Notice that the problem is convex and the optimal value f_{opt} is achieved by running the two-level ADMM until convergence. For the two-level ADMM, we choose $(\tau, \eta, \rho^1) = (0.5, 1.5, \max\{1, \gamma_2, \gamma_3\})$ to start with. In the k-th inner-level ADMM, we terminate our inner updates either the condition (3.14) is satisfied with $\epsilon_i^k = 0.001/k * \|\mathbf{A}\|_*, i = 1, 2, 3$ or the maximum number of ADMM loops 20 is reached. For both algorithms, we choose three different penalty parameters β: 1, 5, and 10. For $\beta = 1$, the multi-block ADMM converges faster than the two-level ADMM while the two-level ADMM shows better convergence results when $\beta = 5$ or 10.

To test the effect of ρ^1, we choose $\beta = 1$ and keep other parameters the same as above. Figure 3.5 plots the convergence results for $\rho^1 = 1, 10, 100, 1000$, and 10000. Except $\rho^1 = 1$, all other four scenarios almost converge with the same speed. It indicates that the two-level ADMM is very robust with respective to the selection of ρ^1.

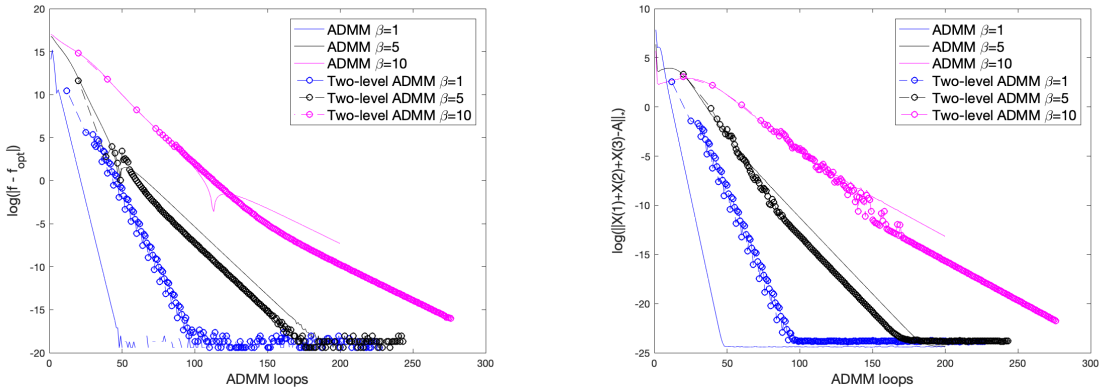


Figure 3.4: Multi-block ADMM and two-level ADMM on RPCA. Left Figure: Comparison on objective value $|f - f_{opt}|$. Right Figure: Comparison on feasibility $\|\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 - \mathbf{A}\|_*$.

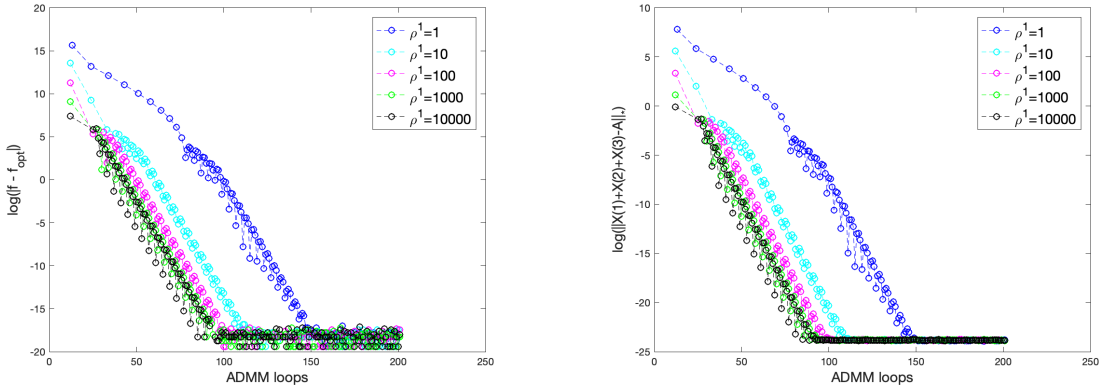


Figure 3.5: The effect of ρ^1. Left Figure: Comparison on objective value $|f - f_{opt}|$. Right Figure: Comparison on feasibility $\|\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 - \mathbf{A}\|_*$.

3.6.3 Compressed sensing problem

We consider the following compressed sensing problem to test the performance of the two-level ADMM:

$$\begin{aligned} \min_{\mathbf{x}} & \sum_{i=1}^{m} \mathcal{P}(x_i) \\ \text{s.t.} & \sum_{i=1}^{m} A_i x_i = b \end{aligned} \tag{3.43}$$

where $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$, $A_i \in \mathbb{R}^n$ and $\mathcal{P}(x)$ represents a penalty function. We consider the following options for the penalty function:

(a) ℓ_1 penalty: $\mathcal{P}(x) = \|x\|_1$;

(b) SCAD penalty:

$$\mathcal{P}(x) = \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda, \\ \dfrac{2\gamma\lambda|x| - x^2 - \lambda^2}{2(\gamma-1)} & \text{if } \lambda < |x| < \gamma\lambda, \\ \dfrac{\lambda^2(\gamma+1)}{2} & \text{if } |x| \geq \gamma\lambda. \end{cases}$$

(c) Smoothed-SCAD penalty:

$$\mathcal{P}(x) = \begin{cases} \lambda(x^2+\epsilon)^{\frac{1}{2}} & \text{if } (x^2+\epsilon)^{\frac{1}{2}} \leq \lambda, \\ \dfrac{2\gamma\lambda(x^2+\epsilon)^{\frac{1}{2}} - x^2 - \lambda^2}{2(\gamma-1)} & \text{if } \lambda < (x^2+\epsilon)^{\frac{1}{2}} < \gamma\lambda, \\ \dfrac{\lambda^2(\gamma+1)}{2} & \text{if } (x^2+\epsilon)^{\frac{1}{2}} \geq \gamma\lambda. \end{cases}$$

Both options (a) and (b) are capable of finding sparse solutions. If we take the ℓ_1 penalty, (3.43) turns out to be the basis pursuit problem in [19]. The SCAD penalty was proposed in [29] to select variables efficiently in high dimension statistics while the function is nonconvex and nonsmooth. To fit into our framework, in option (c), we bypass the nonsmooth problem at $x = 0$ by using a small positive number ϵ to obtain the smooth approximation of the SCAD penalty.

The numerical experiments are performed on some randomly generated data sets of size $(m, n) = (1000, 100)$. The entries of A are generated from i.i.d $\mathcal{N}(0, 1)$ and components of the ground truth x^0 are i.i.d $\mathcal{N}(0, 1)$ with sparsity level 0.1. The remaining variables of x^0 are set to 0 and $b = Ax^0$. For the (smoothed)-SCAD penalty, the parameters are $(\lambda, \gamma, \epsilon) = (2, 4, 1e-3)$.

For the two-level ADMM, we choose $(\tau, \eta, \beta) = (0.5, 1.5, 1)$ to start with. In the k-th inner-level ADMM, we terminate our inner updates either the condition (3.14) is satisfied with $\epsilon_i^k = 0.1/k, i = 1, 2, 3$ or the maximum number of ADMM loops 100 is reached. Here, one ADMM loop represents consecutive updates from x_1 to x_m for one time. The update for z has a closed form solution and the computational effort can be ignored compared with the ADMM loop update. We compare the performance of the Algorithm 3.1 with the two-level ADMM in all three cases in the following figures.

We choose three different options for β: 0.01, 0.1 and 1 in Algorithms 3.1 and 3.2. From Figure 3.6, the two-level ADMM shows comparable convergence results with the multi-block ADMM. We

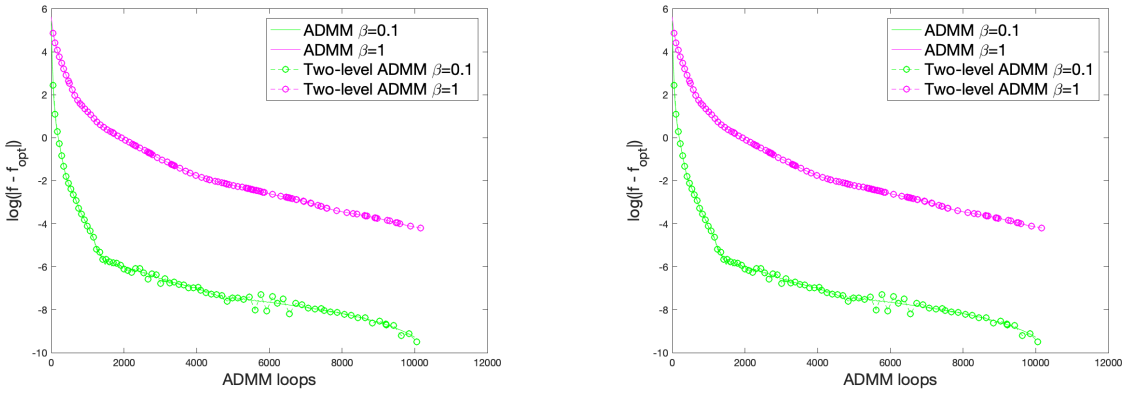


Figure 3.6: Multi-block ADMM and two-level ADMM on ℓ_1 penalty. Left Figure: Comparison on objective value $f - f_{opt}$. Right Figure: Comparison on feasibility $\|Ax - b\|$.

notice that the choice the regularization parameter β in Algorithms 3.1 and 3.2 plays an important role in the convergence speed. It's not a surprise that the Algorithm 3.1 converges for this multi-block problem as it has been shown in [47] that the multi-block ADMM is guaranteed to converge for this particular problem.

For the (smoothed)-SCAD penalty, we select three different options for for β: 0.1, 1 and 10 in Algorithms 3.1 and 3.2. In these two penalties, there is no theoretical guarantee that Algorithm 3.1 will still converge. In Figure 3.7, the two-level ADMM converges nearly the same as the multi-block ADMM in terms of objective values while our algorithm obtains a more stable output regarding the feasibility $\|Ax - b\|$. In Figure 3.8, we see that the multi-block ADMM often oscillates both in function values and constraints. This indicates that the multi-block ADMM may diverge in certain cases. Clearly, Condition 2 is not satisfied in this problem as

$$\mathrm{Im}[A_{1:(m-1)}; b] \nsubseteq \mathrm{Im} A_m.$$

On the other hand, the two-level ADMM shows robust convergence results in both objective values and feasibilities. The above results show that the two-level ADMM can achieve the same convergence speed with multi-block ADMM with a robust convergence guarantee.

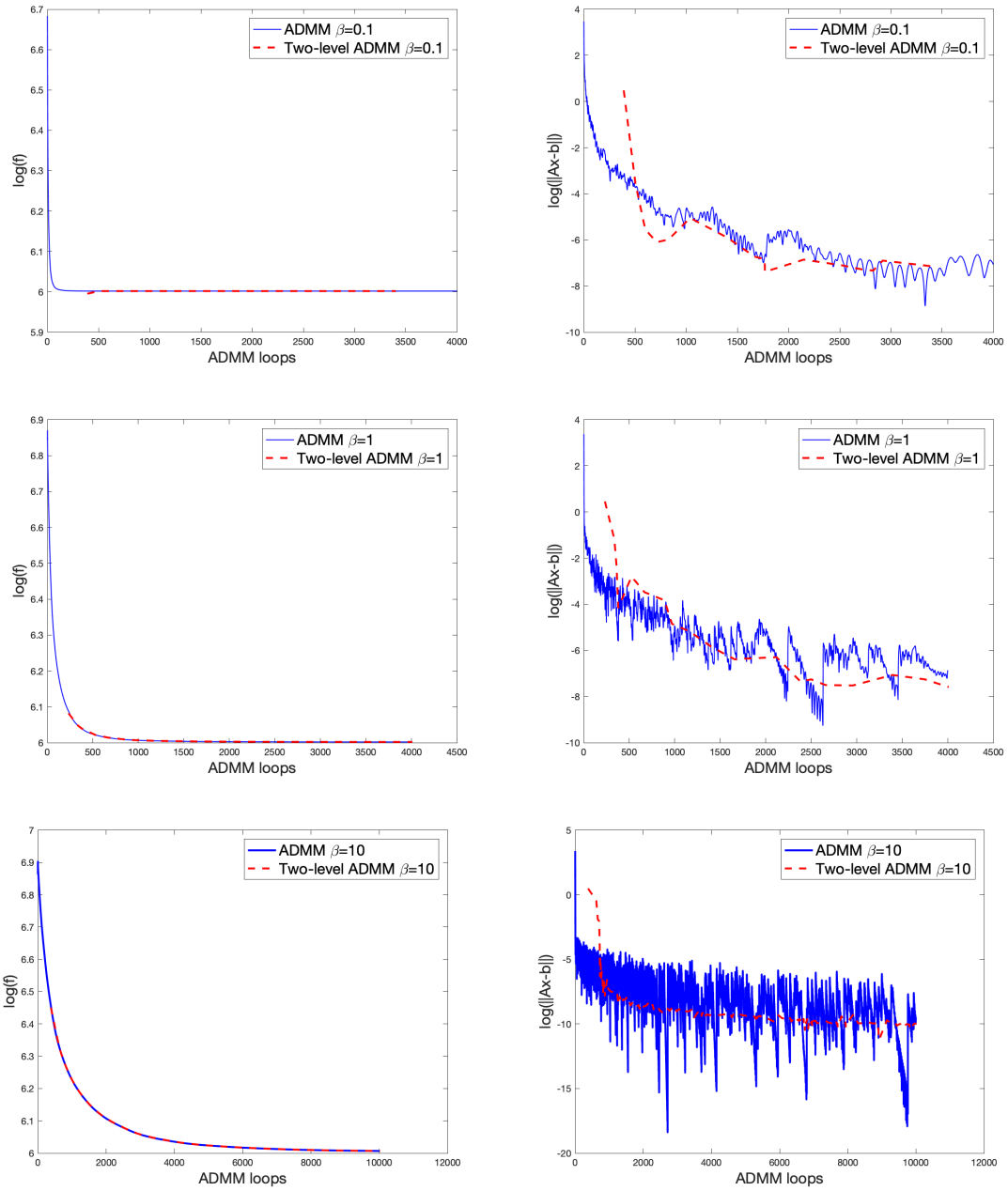


Figure 3.7: Multi-block ADMM and two-level ADMM on SCAD penalty. Left Figures: Comparison on objective value f. Right Figures: Comparison on feasibility $\|Ax - b\|$.

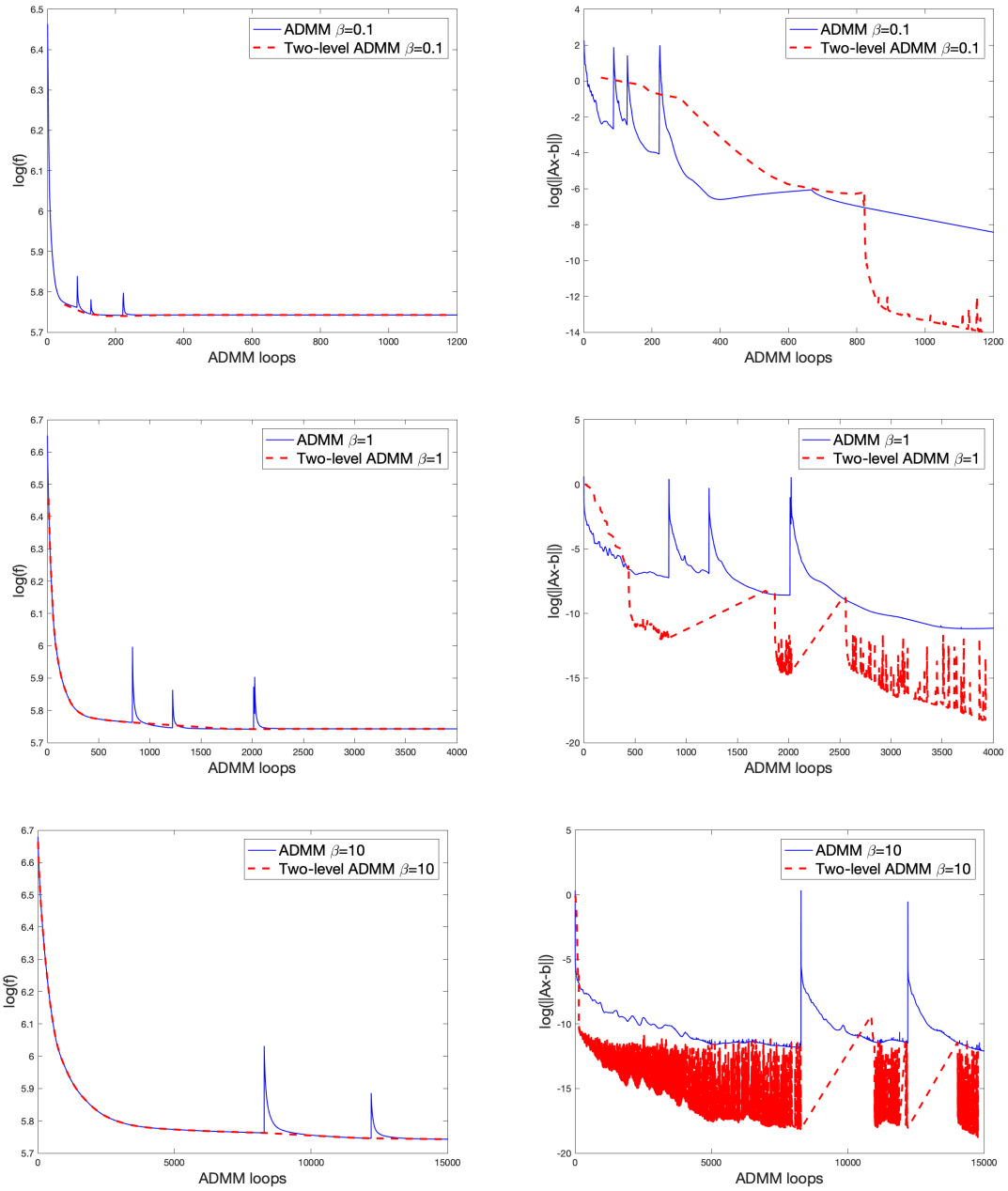


Figure 3.8: Multi-block ADMM and two-level ADMM on the smoothed-SCAD penalty. Left Figures: Comparison on objective value f. Right Figures: Comparison on feasibility $\|Ax - b\|$.

3.7 Future research directions

Here, we discuss a few possible directions for future works based on the two-level ADMM.

(a) From our assumptions, we notice that our convergence results require the compactness of the feasible set. Even though this condition is often satisfied in practical problems, we hope to remove this assumption and replace it with a weaker assumption.

(b) In the numerical part, we keep the inner-level ADMM penalty parameter β fixed in the two-level ADMM while our theoretical analysis implies we need to update β based on ρ. A more sophisticated analysis may help bridge the gap between experiments and theories.

(c) We notice that the two-level ADMM works well in the SCAD penalty which indicates that we may extend the our analysis to a broader class of functions.

CHAPTER 4: A Stochastic Newton Method for Self-Concordant Funtions

4.1 Introduction

We consider the following composite convex minimization problem of a finite sum and a nonsmooth convex regularizer which covers various machine learning and statistics applications [10, 90]:

$$F^\star := \min_{w\in\mathbb{R}^p} \Big\{ F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w) + g(w) \Big\}. \tag{4.1}$$

Here, $f(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w)$ is a convex function that is a finite average of n smooth convex functions $f_i(w)$ and $g(w)$ is a proper, closed, and convex function but possibly nonsmooth. Very often, g is referred to as a regularizer or penalty. While existing methods rely on Lipschitz gradient/Hessian and/or strong convexity of the f or g, we instead exploit the following assumption:

Assumption 4.1. *$f(\cdot)$ is $(M_f, 2)$-generalized self-concordant as defined in Definition 4.1 below.*

This class of functions intersects with Lipschitz gradient/Hessian functions but not a subset of those. Therefore, it covers other problems whose objective functions do not have Lipschitz gradient/Hessian. We will discuss more details about this class of functions in the next section, and give concrete examples.

4.1.1 Motivation and objectives

Although many machine learning problems often require low accurate solutions which are perfectly suitable to solve by first-order methods such as stochastic gradient descent-type methods including variance reduction and dual coordinate descent [2, 22, 85, 109]. However, various applications also require high accurate solutions due to some hard constraints such as feasibility. Example of these hard constraints include nonnegativity constraints in nonnegative matrix factorization, positive semidefiniteness in semidefinite programming, or simplex constraints in portfolio optimization. If we recast problems with these constraints into the setting (4.1), then it is natural to solve them with a high accuracy to likely satisfy these hard constraints. This requirement motivates the use of second-order methods which can achieve high accurate solutions after a few dozens of iterations. Another reason is that second-order methods often require a small number

of iterations, but the per-iteration complexity may be high. A good trade-off between iteration-complexity and per-iteration complexity can make second-order methods win first-order methods in many applications.

In recent years, there has been an emerging trend in developing second-order methods for convex optimization, including (4.1). Among different approaches, subsampling [1, 28, 110] and sketching [76] seem to be the most popular ones. These ideas have been integrated into both pure Newton and quasi-Newton-type methods in different ways. Started from [13], several methods have been proposed to solve different instances of (4.1) such as [1, 8, 28, 76, 83, 84, 110]. The main assumption standing out these works is the **global boundedness of the Hessian** $\mu\mathbb{I} \preceq \nabla^2 f(x) \preceq L\mathbb{I}$ for all x in a given sublevel set in some sense. Unfortunately, this assumption excludes some important applications such as problems involving self-concordant functions [71], reciprocal functions, and exponential objectives [66]. Moreover, in some cases, if this assumption is satisfies, its condition number may still be very large if the bounds are conservative. Hence, the universal learning rate computed from this bound can be too small for practical purpose, which leads to a poor performance of the algorithm. Note that Newton-type methods often have a fast local convergence, but their global convergence rates remains sublinear. Existing methods often rely on linesearch procedures or trust-region strategies to guarantee a descent property which are not well-understood in stochastic variants. Therefore, existing analysis often assumes the above universal boundedness assumption to derive a learning rate.

Our goal is to exploit a recent concept called "generalized self-concordance" in [93] to develop a class of subsampled proximal-Newton-type methods for solving (4.1) under Assumption 4.1. We show that our assumption intersects with the boundedness assumption, and hence covers other applications that cannot be solved by existing subsampled Newton-type methods (at least in terms of theoretical guarantees). While our theory can be extended to a more general class of functions studied in [93], we will focus on the case of generalized self-concordant with $\nu = 2$ for sake of presentation.

4.1.2 Contribution

Our contribution can be summarized as follows:

(a) We exploit a new concept called "generalized self-concordance" for smooth convex functions

to develop a novel inexact subsampled proximal-Newton algorithm to solve (4.1) under Assumption 4.1. One of our main contribution is an explicit learning rate (or step-size) without any globalization procedure such as linesearch or trust-region.

(b) Our second main contribution is a new analysis of both global convergence and local linear-quadratic convergence of our method under Assumption 4.1. We analyze local convergence rate of both variants: damped-step and full-step schemes.

(c) We also analyze convergence results for the case where both Hessian and the gradients are subsampled. We prove both global convergence and local linear-quadratic convergence rate of this variant.

(d) To scale up to high-dimensional problems, we propose to combine with coordinate descent strategies to obtain new variants. Due to space limit, we briefly present one variant in Subsection 4.3.7. The full algorithm and its convergence analysis can be found in Supplementary Document.

(e) We provide two representative examples to illustrate two cases. The first example is a sparse logistic regression to demonstrate our methods over existing methods. We show that our algorithm is much faster than exact proximal-Newton methods and comparable with the current state-of-the-art algorithms on several real datasets. The second problem is a sparse Poisson regression which shows that our methods are applicable to other models where existing methods do not have theoretical guarantees due to the lack of bounds on Hessian.

As we stated, we only focus on a particular class of "generalized self-concordant functions" studied in [93] with $\nu = 2$. However, our analysis can be easily extended to cover more broader class of convex functions without much effort.

4.1.3 Notations and terminologies

Given a proper, closed, and convex function $f : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, we use $\mathrm{dom}(f)$ and ∂f to denote its domain and subdifferential, respectively. We use $\mathcal{C}^3(\mathrm{dom} f)$ to denote the class of three times continuously differentiable functions on its open domain $\mathrm{dom} f$. If $\nabla^2 f(x) \succ 0$ at a given $x \in \mathrm{dom} f$, then we define a local norm $\|u\|_x := \langle \nabla^2 f(x)u, u\rangle^{1/2}$ as a weighted norm of u with respect to $\nabla^2 f(x)$. The corresponding dual norm $\|v\|_x^*$, is defined as $\|v\|_x^* := \max\{\langle v, u\rangle \mid \|u\|_x \leq 1\} = \left\langle \nabla^2 f(x)^{-1} v, v\right\rangle^{1/2}$ for $v \in \mathbb{R}^p$. For any $\mathbf{A} = (\mathbf{A}_1, \cdots, \mathbf{A}_n) \in \mathbb{R}^{p\times n}$, we use $\|\mathbf{A}\|_2$ to denote its

spectral norm and $\|\mathbf{A}\|_F := \left(\sum_{j=1}^n \|\mathbf{A}_j\|_2^2\right)^{1/2}$ to define its Frobenius norm. The *stable rank* of a nonzero matrix $\mathbf{A}$ is $\mathsf{sr}(\mathbf{A}) \equiv \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_2^2}$, where $1 \leq \mathsf{sr}(\mathbf{A}) \leq \mathrm{rank}(\mathbf{A})$. We say that a differentiable function f is *L-smooth* or $f \in \mathcal{F}_L$ if

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y \in \mathrm{dom}(f).$$

Throughout the paper, we make use of the following standard assumptions:

A.1 **Locally Strong Regularity:** $f(w)$ is locally strongly convex, *i.e.*,

$$\sigma_{\min}(w^\star) = \lambda_{\min}(\nabla^2 f(w^\star)) > 0,$$

where $w^\star$ is any optimal solution of (4.1).

A.2 **Hessian Decomposition:** For each f_i in (4.1), define $\nabla^2 f_i(w) := \mathbf{A}_i(w)\mathbf{A}_i(w)^\top$ where $\mathbf{A}_i(w) \in \mathbb{R}^p$ and denote $\mathbf{A}(w) = (\mathbf{A}_1(w), \cdots, \mathbf{A}_n(w)) \in \mathbb{R}^{p\times n}$. Note that $\nabla^2 f(w) = \mathbf{A}(w)\mathbf{A}(w)^\top$. Since $\nabla^2 f(w) \succeq 0$, this assumption is very natural.

Given a sequence $\{a_t\} \subset \mathbb{R}$, we say that $\{a_t\}$ linear-quadratically converges to zero if there exists $t_0 \geq 0$ such that for all $t \geq t_0$, we have $|a_{t+1}| \leq C_1 a_t^2 + C_0 |a_t|$ for some $C_1 > 0$ and $C_0 \in (0, 1)$.

Paper organization: The rest of this chapter is organized as follows. Section 4.2 introduces the definition of *generalized self-concordant* functions and provides key properties of this function class. Some representative examples are also given. Section 4.3 proposes an inexact subsampled proximal-Newton method, its variants, and investigates its convergence guarantees. Numerical experiments are presented in Section 4.4 to demonstrate the effectiveness and robustness of our algorithms.

4.2 Background

Let $f : \mathbb{R}^p \to \mathbb{R}$ be a $\mathcal{C}^3(\mathrm{dom} f)$ smooth and convex function with open $\mathrm{dom}(f)$. Given $\nabla^2 f$ the Hessian of f, $x \in \mathrm{dom}(f)$, and $u, v \in \mathbb{R}^p$, we consider the function $\psi(t) := \langle \nabla^2 f(x + tv)u, u\rangle$. Then, it is obvious to show that $\psi'(t) := \langle \nabla^3 f(x + tv)[v]u, u\rangle$ for $t \in \mathbb{R}$ such that $x + tv \in \mathrm{dom} f$, where $\nabla^3 f$ is the third-order derivative of f. It is clear that $\psi(0) = \langle \nabla^2 f(x)u, u\rangle = \|u\|_x^2$.

Definition 4.1 ([93]). *A $\mathcal{C}^3$-convex function f is said to be an (M_f, ν)-generalized self-concordant function of order $2 \leq \nu \leq 3$ if for any $w \in \mathbb{R}^p$ and $u, v \in \mathbb{R}^p$, then*

$$|\langle \nabla^3 f(w)[v]u, u\rangle| \leq M_f \|u\|_w^2 \|v\|_w^{\nu-2} \|v\|_2^{3-\nu}. \tag{4.2}$$

Note that if $\nu = 3$ and $u = v$, (4.2) reduces to $|\langle \nabla^3 f(w)[u]u, u\rangle| \leq M_f \|u\|_w^3$, Definition 4.1 defines the standard self-concordant concept introduced in [70, 71]. In this chapter, we consider the composite convex optimization problem (4.1) in which f is $(M_f, 2)$-*generalized self-concordant*. The following proposition shows that if a *generalized self-concordant* function has a Lipschitz gradient, then it can be cast into the special case $\nu = 2$.

Proposition 4.1 ([93])**.** *Let f be (M_f, ν)-generalized self-concordant with $\nu \in (2, 3]$. If, in addition, f is L_f-smooth, then f is also $(\hat{M}_f, 2)$-generalized self-concordant with $\hat{M}_f := M_f L_f^{\frac{\nu}{2}-1}$.*

Now, we provide some representative empirical loss functions using in regression and classification that are $(M_f, 2)$-*generalized self-concordant* . In such problems, we are given n training examples $\{(x_1, y_1), \cdots (x_n, y_n)\}$ where each $x_i \in \mathbb{R}^p$ is the feature vector of example i, and each $y_i \in \mathbb{R}$ is the label of example i.

(a) Poisson regression: We consider the following ℓ_1-regularized Poisson regression problem, e.g., in [51]

$$\min_{w \in \mathbb{R}^p} \left\{ F(w) = f(w) + \lambda \|w\|_1 \right\}, \tag{4.3}$$

where $f(x) := \frac{1}{n} \sum_{i=1}^n (y_i e^{-0.5 x_i^\top w} + e^{0.5 x_i^\top w})$, $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{Z}_+$, and $\lambda \in \mathbb{R}_+$. The following lemma shows that f is $(M_f, 2)$-*generalized self-concordant*.

Lemma 4.1. *The function f defined in* (4.3) *is $(M_f, 2)$-generalized self-concordant with $M_f := \frac{1}{2} \max_{1 \leq i \leq n} \|x_i\|_2$.*

(b) Binary classification: We consider the following ℓ_1-regularized empirical risk problem:

$$\min_{w \in \mathbb{R}^p} \Big\{ F(w) := \frac{1}{n} \sum_{i=1}^n \varphi(y_i w^\top x_i) + g(w) \Big\}, \tag{4.4}$$

where $x_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$, $\lambda \in \mathbb{R}_+$, $f(x) := \frac{1}{n} \sum_{i=1}^n \varphi(y_i w^\top x_i)$ is a empirical loss function, and g is a regularizer, e.g., $g(w) := \lambda \|w\|_1$ for sparse settings, or $g(w) = \frac{\lambda}{2} \|w\|_2^2$ for ridge settings. Commonly used loss functions include:

- Logistic loss: $\varphi(s) := \log(1 + e^{-s})$

- Hinge loss: $\varphi(s) := \max\{0, 1-s\}$ and its smoothed version

$$\varphi_\gamma(s) := \gamma \ln\left(\frac{e^{(1-s)/\gamma} + e^{-(1-s)/\gamma}}{2}\right) + \tfrac{1}{2}(1-s)$$

  for some smoothness parameter $\gamma > 0$.

The following lemma shows that these loss functions are $(M_f, 2)$-*generalized self-concordant*.

Lemma 4.2. *For logistic regression, the empirical loss f is $(M_f, 2)$-generalized self-concordant with $M_f := \max_{1\le i\le n} \|x_i\|_2$. For smoothed hinge regression, the empirical loss f is $(M_f, 2)$-generalized self-concordant with $M_f := \frac{2}{\gamma}\max_{1\le i\le n} \|x_i\|_2$.*

The proof for previous lemmas can be found in Supplementary Document. The following theorem guarantees that problem (4.1) has a unique solution $w^\star$.

Theorem 4.1 (Theorem 4 in [93])**.** *Suppose that the function f of (4.1) is $(M_f, 2)$-generalized self-concordant with $M_f > 0$. Denote $\sigma_{\min}(w) := \lambda_{\min}(\nabla^2 f(w))$ and $\lambda(w) := \|\nabla f(w) + v\|_w^*$ for some $v \in \partial g(w)$. Suppose further that there exists $x \in \mathrm{dom}(F)$ such that $\sigma_{\min}(w) > 0$ and $\lambda(w) < M_f^{-1}\sqrt{\sigma_{\min}(w)}$. Then, problem (4.1) has a unique solution $w^\star \in \mathrm{dom}(F)$.*

4.3 The inexact subsampled proximal-Newton algorithm

We develop an inexact subsampled proximal-Newton (PN) method, and establish its global and local convergence. Then, we present its variants.

4.3.1 Derivation of the algorithm

At each iteration $t \ge 0$ of our method for solving (4.1), we compute an inexact proximal-Newton direction by approximately solving the following subproblem

$$v_t \approx \operatorname*{argmin}_v \Big\{ \widetilde{\nabla} f(w_t)^\top v + \tfrac{1}{2} v^\top \mathbf{H}_t v + g(w_t + v) \Big\}, \tag{4.5}$$

where $\widetilde{\nabla} f(w_t)$ is a stochastic approximation of $\nabla f(w_t)$ at w_t and $\mathbf{H}_t \succ 0$ is a subsampled approximation to $\nabla^2 f(w_t)$ which satisfies the following condition:

$$\|\mathbf{H}_t - \nabla^2 f(w_t)\|_2 \le \beta_t \|\nabla^2 f(w_t)\|_2. \tag{C1}$$

Note that (C1) is a common guarantee for matrix approximation problems [110, 111].

In our algorithm, the following criterion will be used for accepting a vector v_t as an *inexact proximal-Newton step* at w_t: there exists an error r_t such that

$$\begin{aligned} r_t &\in \widetilde{\nabla} f(w_t) + \mathbf{H}_t v_t + \partial g(w_t + v_t), \\ \|r_t\|_{\mathbf{H}_t}^* &\leq (1-\theta_t)\,\|v_t\|_{\mathbf{H}_t} \end{aligned} \tag{4.6}$$

for some $\theta_t \in (0.9, 1]$. One can view $1-\theta_t$ as a bound on the relative error for solving the subproblem (4.5). If $\theta_t = 1$, then we have an exact solution of (4.5). Once v_t is computed, we define

$$\lambda_t := \|v_t\|_{w_t}, \quad d_2(v_t) := M_f \|v_t\|_2 \tag{4.7}$$

and

$$\gamma_t = (1+\beta_t)(1-\theta_t) + \beta_t.$$

Then, we update the new iteration using the following damped step-size η_t:

$$w_{t+1} = w_t + \eta_t v_t, \text{ with } \eta_t := \tfrac{\ln(1+(1-\gamma_t)d_2(v_t))}{d_2(v_t)}. \tag{4.8}$$

For simplicity of analysis, we divide it into two cases: Exact gradient $\widetilde{\nabla} f(w_t) = \nabla f(w_t)$, and subsampled-gradient $\widetilde{\nabla} f(w_t) \approx \nabla f(w_t)$.

4.3.2 Convergence analysis: Exact gradient

We now present convergence results of the damped-step Newton scheme (4.8) in the following theorem whose proof can be found in Supplementary Document.

Theorem 4.2. *Let $\{w_t\}$ be the sequence generated by the scheme* (4.8) *with exact gradient estimate $\widetilde{\nabla} f(w_t) = \nabla f(w_t)$. If we solve the subproblem* (4.5) *until* (4.6) *and* (C1) *are met, then the following statements hold:*

(i) *The step-size η_t in* (4.8) *guarantees:*

$$F(w_{t+1}) \leq F(w_t) - \Delta_t, \tag{4.9}$$

where $\Delta_t := (1-\gamma_t)\eta_t\lambda_t^2 - \omega_2(\eta_t d_2(v_t))\lambda_t^2 > 0$ with $\omega_2(\tau) := \frac{e^\tau - \tau - 1}{\tau^2}$.

(ii) *There exists a neighborhood $\mathcal{N}(w^\star)$ of the solution $w^\star$ of* (4.1) *such that if we choose $w_0 \in$*

$\mathcal{N}(w^\star) \cap \mathrm{dom}(F)$ *and assume* $\sup_t \{\gamma_t\} \leq 0.2$*, then* $\{w_t\}$ *converges to* $w^\star$ *at a linear-quadratic rate.*

Next, we study a full-step proximal-Newton scheme derived from (4.8) by letting the step-size $\eta_t = 1$ for all $t \geq 0$. Let $\underline{\sigma}_t$ be the smallest eigenvalue of $\nabla^2 f(w_t)$. Since $\nabla^2 f(w_t) \succ 0$, we have $\underline{\sigma}_t > 0$. The following theorem shows a local linear-quadratic convergence of the *full-step* inexact proximal-Newton scheme, whose proof can be found in Supplementary Document.

Theorem 4.3. *Let* $\{w_t\}$ *be the sequence generated by the full-step inexact proximal-Newton scheme with* $\eta_t = 1$*. Suppose that we solve* (4.5) *until* (4.6) *and* (C1) *are met. If the starting point* w_0 *satisfies* $\frac{\lambda_0}{\sqrt{\underline{\sigma}_0}} \leq \frac{1}{4M_f}$ *and* $\sup_t \{\gamma_t\} \leq 0.2$*, then both sequences* $\left\{\frac{\lambda_t}{\sqrt{\underline{\sigma}_t}}\right\}$ *and* $\{d_2(v_t)\}$ *decrease and linear-quadratically converge to zero. Consequently,* $\{\|w_t - w^\star\|_2\}$ *also locally converges to zero at a linear-quadratic rate.*

4.3.3 The full algorithm

Combining the results of Theorem 4.2 and Theorem 4.3, we can design a new *inexact subsampled proximal-Newton algorithm* for solving (4.1) under the structural assumption, Assumption 4.1, as follows:

— *Phase 1*: Starting from an arbitrary initial point $w_0 \in \mathrm{dom}(F)$, we perform the damped-step proximal-Newton scheme (4.8) until the condition in Theorem 4.3 is satisfied.

— *Phase 2* (**optional**): Using the output w_t of Phase 1 as an initial point for the full-step proximal-Newton scheme (4.8) with $\eta_t = 1$, and perform this scheme until it converges.

Note that *Phase 2* is only **optional**. We can only run *Phase 1* until we achieve a desired solution.

Algorithm 4.1 (Inexact subsampled proximal-Newton (iSSPN) algorithm for solving (4.1) under Assumption 4.1)

Input: $w_0 \in \mathrm{dom}(F)$, a number of iterations T, $\theta_t \in (0.9, 1], \beta_t \in (0, 0.08]$, sampling size c.

Output: an approximation w_T of $w^\star$ of (4.1).

for $t = 0, 1, \cdots, T-1$ **do**

  Construct an approximation $\widetilde{\nabla} f(w_t)$ of $\nabla f(w_t)$.

  Construct the sampling distribution $\{p_i\}_{i=1}^n$ that is independent of $\widetilde{\nabla} f(w_t)$.

  for $i = 1, \cdots, n$ **do**

    Set $q_i = \min\{c \cdot p_i, 1\}$ and compute

$$\tilde{\mathbf{A}}_i(w_t) := \begin{cases} \mathbf{A}_i(w_t)/\sqrt{q_i}, & \text{with probability } q_i, \\ 0, & \text{with probability } 1 - q_i \end{cases}$$

  Construct $\mathbf{H}_t := \frac{1}{n}\sum_{i=1}^n \tilde{\mathbf{A}}_i(w_t)\tilde{\mathbf{A}}_i(w_t)^\top$.

  Solve (4.5) approximately until (4.6) is met.

  Compute $d_2(v_t) := M_f \|v_t\|_2$ and a learning rate:

$$\eta_t := \begin{cases} \frac{\ln(1+(1-\gamma_t)d_2(v_t))}{d_2(v_t)}, & \text{if Phase 1 is used} \\ 1, & \text{if Phase 2 is used.} \end{cases} \tag{4.10}$$

  Update $w_{t+1} = w_t + \eta_t v_t$

return w_T.

Conceptually, the two-phase option of Algorithm 4.1 requires the smallest eigenvalue of $\nabla^2 f(w_t)$ to terminate Phase 1. However, switching from Phase 1 to Phase 2 can be done automatically allowing some tolerance in the step-size η_t or equivalently $d_2(v_t)$. Indeed, the step-size η_t given in (4.10) converges to $1-\gamma_t$ as $t \to \infty$. Hence, when η_t is close enough to $1-\gamma_t$, we can automatically set it to 1 and remove the computation of λ_t to reduce computational time.

4.3.4 Inexactness of subproblems

To check the inexact stopping criterion in (4.6), we need a tractable formulation of r_t. In the experiment, we use the FISTA algorithm [7] to minimize the function (4.5) in the subproblems. At

the k-th iteration of FISTA, it follows the update

$$v^k = \text{prox}_{\alpha g}\left(w + u - \alpha(\nabla f(w) + \mathbf{H}_t u)\right) - w, \tag{4.11}$$

where u is related to v^{k-1} and v^{k-2} as following:

$$u = v^{k-1} + \left(\tfrac{k-2}{k+1}\right)(v^{k-1} - v^{k-2}).$$

From the definition of the proximal operator $\text{prox}_{\alpha g}$, the following inclusion holds:

$$\tfrac{1}{\alpha}(u - v^k) \in \nabla f(w) + \mathbf{H}_t u + \partial g(w + v^k).$$

This implies that the vector

$$r = \tfrac{1}{\alpha}(u - v^k) + \mathbf{H}_t(v^k - u) = \left(\tfrac{1}{\alpha}\mathbb{I} - \mathbf{H}_t\right)(u - v^k)$$

satisfies $r \in \nabla f(w) + \mathbf{H}_t v^k + \partial g(w + v^k)$. In our experiment, $r = (\frac{1}{\alpha}\mathbb{I} - \mathbf{H}_t)(u - v^k)$ was used to check whether the condition (4.6) has been satisfied.

Notice that the main computational efforts relate to the following Hessian-vector product

$$\mathbf{H}_t u = \frac{1}{n}\sum_{i=1}^{n} \tilde{\mathbf{A}}_i(w_t)\tilde{\mathbf{A}}_i(w_t)^\top u_t.$$

The above update suggests that the full Hessian matrix is not required in practice.

4.3.5 Sufficient sampling size

The following theorem [46, 110] shows the approximation error bound for the Gram matrix in different sampling themes. The results guarantee that (C1) can be satisfied with high probability with a certain sampling size. We give a short proof in Supplementary Document.

Theorem 4.4. *Given any $\beta_t \in (0, 1)$, the following statements hold:*

(i) (Uniform sampling) *Let $\mathbf{H}_t$ be constructed by Algorithm 4.1 with the probability $p_i := \frac{1}{n}$, $(i = 1, \cdots, n)$. Then, if $c \geq 4n \cdot \frac{\max_i \|\mathbf{A}_i(w_t)\|^2}{\|\mathbf{A}(w_t)\|_2^2} \cdot \log\left(\frac{p}{\delta}\right) \cdot \frac{1}{\beta_t^2}$, then, with probability at least $1 - \delta$,* (C1) *holds.*

(ii) (Non-uniform sampling) *Let $\mathbf{H}_t$ be constructed by Algorithm* (4.1) *with the probability $p_i :=$*

$\frac{r_i}{\sum_{j=1}^{n} r_j}$, *where* $r_i = \|\mathbf{A}_i(w_t)\|^2, i = 1, \cdots, n$. *Then, if* $c \geq 4\mathsf{sr}(\mathbf{A}(w_t)) \cdot \log\left(\frac{p}{\delta}\right) \cdot \frac{1}{\beta_t^2}$, *then, with probability at least* $1 - \delta$, *condition* (C1) *holds.*

From Theorem 4.4, we might need $\Omega(n)$ samples in the extreme case when we implement the uniform sampling method. However, the non-uniform sampling method only requires $\mathcal{O}(p \log p)$ samples which is independent of the number of samples n. In the context of Theorem 4.4, the convergence results in Theorems 4.2 and 4.3 are guaranteed with high probability, i.e., at least $1 - \delta$.

4.3.6 Convergence analysis: Subsampled-gradient

In the second case, we subsample both gradient and Hessian. Let $\mathcal{S}$ and $|\mathcal{S}|$ denote a random sample collection and its cardinality respectively. Let

$$\widetilde{\nabla} f(w) := \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla f_j(w) \tag{4.12}$$

be the sub-sampled gradient of $\nabla f(w)$, which is independent of the subsample in the Hessian. We require the approximate gradient $\widetilde{\nabla} f(w_t)$ to satisfy the following condition:

$$\|\nabla f(w_t) - \widetilde{\nabla} f(w_t)\|_{w_t}^* \leq \xi_t \lambda_t \text{ and } \xi_t \leq 0.05. \tag{C2}$$

Here, we provide both global and local convergence guarantee of this variant in the following theorem whose proof is given in Supplementary Document.

Theorem 4.5. *Let* $\{w_t\}$ *be the sequence generated by* (4.8) *with subsampled-gradient estimate* (4.12). *If we solve subproblem* (4.5) *until* (4.6), (C1), *and* (C2) *are met, then the following statements hold:*

(i) *The step-size* η_t *in* (4.5) *guarantees:*

$$F(w_{t+1}) \leq F(w_t) - \tilde{\Delta}_t, \tag{4.13}$$

where $\tilde{\Delta}_t := \left(\eta_t(1 - \tilde{\gamma}_t) - \eta_t^2 \omega_2(\eta_t d_2(v_t))\right) \lambda_t^2 > 0$ *with* $\tilde{\gamma}_t := \gamma_t + \xi_t$.

(ii) *There exists a neighborhood* $\mathcal{N}(w^\star)$ *of* $w^\star$ *of* (4.1) *such that if we initialize at* $w_0 \in \mathcal{N}(w^\star) \cap \mathrm{dom}(F)$ *and assume* $\sup_t \{\tilde{\gamma}_t\} \leq 0.25$, *then* $\{w_t\}$ *converges to* $w^\star$ *at a linear-quadratic rate.*

(iii) *Let* $\{w_t\}$ *be the sequence generated by the full-step iSSPN scheme by setting the step-size*

$\eta_t = 1$. *If the starting point* w_0 *satisfies* $\frac{\lambda_0}{\sqrt{\varpi_0}} \leq \frac{1}{5M_f}$ *and* $\sup_t \{\tilde{\gamma}_t\} \leq 0.25$, *then both sequences* $\left\{\frac{\lambda_t}{\sqrt{\varpi_t}}\right\}$ *and* $\{d_2(v_t)\}$ *decrease and linear-quadratically converge to zero. Consequently,* $\{\|w_t - w^\star\|\}$ *also locally converges to zero at a linear-quadratic rate.*

The following lemma probabilistically controls the error for the approximation $\widetilde{\nabla} f(w)$ of $\nabla f(w)$.

Proposition 4.2. *For a given* $w_t \in \mathrm{dom}(F)$ *satisfying* $\|w_t - w^\star\| \leq \frac{1}{2M_f}$, *let* $\|\nabla f_i(w_t)\| \leq G(w_t), i = 1, \cdots, n$. *For any* $0 < \xi_t < 1$ *and* $0 < \delta < 1$, *if* $|\mathcal{S}| \geq \frac{e^{0.25} G(w_t)^2}{\sigma_{\min}(w^\star)\xi_t^2 \lambda_t^2}\left(1 + \sqrt{8\ln\left(\frac{1}{\delta}\right)}\right)^2$, *then* $\widetilde{\nabla} f(w_t)$ *defined by* (4.12) *satisfies*

$$\Pr\left(\|\nabla f(w_t) - \widetilde{\nabla} f(w_t)\|_{w_t}^* \leq \xi_t \lambda_t\right) \geq 1 - \delta. \tag{4.14}$$

Compared with the result in [84] which associates with the local conditional number, the sample size needed in the above proposition only relies on the local regularity. As λ_t decreases to 0, the above bound shows that estimation of the gradient must be done progressively and more accurately while the sample size needed for Hessian approximation can remain unchanged. This is in line with the common knowledge whereas w_t gets closer to the optimal solution $w^\star$, the accuracy of gradient estimation is more significant than that of Hessian.

4.3.7 Block-coordinate iSSPN variant

When the problem dimension p is large, the computational efforts in each iteration of iSSPN is relatively high. In this case, we propose to combine our iSSPN scheme with a block coordinate descent strategy. Let $w = (w^1, \cdots, w^k) \in \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_k}$, where each w_i denotes a subvector of dimension N_i, form a partition of the components w and $\sum_{i=1}^k N_i = p$. Given the current iterate w_t, we randomly choose $\iota \in [k]$ and approximately solve the following subproblem:

$$v_t^\iota \approx \operatorname*{argmin}_{v^\iota} \left\{\widetilde{\nabla}_\iota f(w_t)^\top v^\iota + \tfrac{1}{2} v^{\iota\top} (\mathbf{H}_t)_{\iota\iota} v^\iota + g_\iota(w_t^\iota + v^\iota)\right\} \tag{4.15}$$

which satisfies

$$\begin{aligned} r_t^\iota &\in \widetilde{\nabla}_\iota f(w_t) + (\mathbf{H}_t)_{\iota\iota} v_t^\iota + \partial g(w_t^\iota + v_t^\iota) \\ \|r_t^\iota\|_{(\mathbf{H}_t)_{\iota\iota}}^* &\leq (1 - \theta_t)\, \|v_t^\iota\|_{(\mathbf{H}_t)_{\iota\iota}}, \end{aligned} \tag{4.16}$$

and $\widetilde{\nabla}_\iota f(w_t)$ and $(\mathbf{H}_t)_{\iota\iota}$ are respectively the subvector and submatrix of $\widetilde{\nabla} f(w_t)$ and $\mathbf{H}_t$ corresponding to w^ι. Let us assume that we choose the block coordinate $\iota \in [k]$ with a probability $\bar{p}_i > 0$ for $i = 1, \cdots, k$. That is for $i = 1, \cdots, k$:

$$\Pr(\iota = i) = \bar{p}_i > 0, \text{ with } \sum_{i=1}^{k} \bar{p}_i = 1. \tag{4.17}$$

The full algorithm is presented as in Algorithm 4.2.

Algorithm 4.2 (Block-coordinate iSSPN variant)

Input: $w_0 \in \mathrm{dom}(F)$, number of epochs T, $\theta_t \in (0.9, 1], \beta_t \in (0, 0.08]$, sampling size c.

Output: an approximation w_T of the true solution $w^\star$ of (4.1).

for $t = 0, 1, \cdots, T-1$ **do**

  Randomly choose $\iota \in [k]$ with probability $\bar{p}_\iota$.

  Construct $\widetilde{\nabla}_\iota f(w_t)$ as an approximation of $\nabla f_\iota(w_t)$.

  Construct the sampling distribution $\{p_i\}_{i=1}^n$.

  for $i = 1, \cdots, n$ **do**

    Set $q_i = \min\{c \cdot p_i, 1\}$ and compute

$$\tilde{\mathbf{A}}_i^\iota(w_t) = \begin{cases} \mathbf{A}_i^\iota(w_t)/\sqrt{q_i}, & \text{with probability } q_i, \\ 0, & \text{with probability } 1 - q_i \end{cases}$$

  Construct $(\mathbf{H}_t)_{\iota\iota} := \frac{1}{n}\sum_{i=1}^n \tilde{\mathbf{A}}_i^\iota(w_t)\tilde{\mathbf{A}}_i^\iota(w_t)^\top$.

  Solve the subproblem (4.15) approximately until satisfying criterion (4.16).

  Compute $d_2(v_t^\iota) = M_f||v_t^\iota||_2$ and choose an adaptive step-size (i.e., learning rate):

$$\eta_t^\iota = \begin{cases} \frac{\ln(1+(1-\gamma_t)d_2(v_t^\iota))}{d_2(v_t^\iota)}, & \text{if Phase 1 is used} \\ 1, & \text{if Phase 2 is used.} \end{cases} \tag{4.18}$$

  Update

$$w_{t+1}^j = \begin{cases} w_t^j + \eta_t^j v_t^j, & j = \iota \\ w_t^j, & j \neq \iota \end{cases}$$

return w_T.

The following lemma shows a descent property of this block-coordinate iSSPN variant,

Lemma 4.3. *Let $\{w_t\}$ be generated by the block-coordinate iSSPN variant. Then*

$$\mathbb{E}_\iota[F(w_{t+1})] \leq F(w_t) - \sum_{i=1}^k \bar{p}_i \Delta_t^i \tag{4.19}$$

where $\Delta_t^i := ((1-\gamma_t)\eta_t^i - \omega_2(\eta_t^i d_2(v_t^i)))(\lambda_t^i)^2 > 0$ *with* $\omega_2(\tau) := \frac{e^\tau - \tau - 1}{\tau^2}$.

Proof. Recall that $\iota \in [k]$ is randomly chosen at iteration t with probability $\bar{p}_\iota$. Since f is an $(M_f, 2)$-*generalized self-concordant* function, it is not hard to observe that

$$f(w_t^1, \ldots, w_t^{\iota-1}, z, w_t^{\iota+1}, \ldots, w_t^k)$$

is also an $(M_f, 2)$-*generalized self-concordant* function of z. In view of this and the proof of Theorem 4.2, one can obtain that

$$F(w_{t+1}) \leq F(w_t) - \Delta_t^\iota.$$

Taking expectation with respect to ι, one has

$$\mathbb{E}_\iota[F(w_{t+1})] \leq F(w_t) - \sum_{i=1}^k \bar{p}_i \Delta_t^i. \tag{4.20}$$

This completes the proof. □

This lemma is key to analyze global convergence of our block-coordinate iSSPN variant. Its global convergence guarantee is very similar to Theorem 4.5 and we skip here without proof. We believe that by utilizing the same technique as in [62], one can also achieve a local linear convergence rate of our block-coordinate iSSPN variant.

4.4 Numerical experiments

We provide two representative numerical examples to demonstrate the performance of our algorithms compared to existing state-of-the-art methods.

4.4.1 Sparse Logistic Regression

Given a training dataset $(x_1, y_1), \cdots, (x_n, y_n)$ where $x_i \in \mathbb{R}^p$ and $y_i \in \{+1, -1\}$, we use Algorithm 4.1 to solve the following ℓ_1-regularized logistic regression problem:

$$\min_{w \in \mathbb{R}^p} \Big\{ F(w) = \frac{1}{n} \sum_{i=1}^n \log\left(1 + e^{-y_i w^\top x_i}\right) + \lambda \|w\|_1 \Big\}. \tag{4.21}$$

Clearly, F is non-strongly convex in general. We confirm our theoretical results using several publicly available data sets: (1) the `W8a` dataset (49749 examples and 300 features), (2) the `Adult`

dataset (32,561 examples and 123 feature), (3) the `Covtype` dataset (581,012 examples and 54 feature), and (4) the `MNIST` dataset with 4th and 9th classes (100,000 examples and 784 feature). Each example in these data sets has been normalized so that $\|x_i\|_2 = 1$ for all $i = 1, \cdots, n$. We consider two different values for the regularization parameter $\lambda \in \left\{10^{-4}, 10^{-5}\right\}$ as often used in existing literature.

We implement our iSSPN algorithm, Algorithm 4.1, and compare it with the following methods:

- Proximal-Newton [93]: this is the standard proximal-Newton method under *generalized self-concordance*.
- OWLQN [3]: this is a popular limited-memory quasi-Newton method which can handle the ℓ_1-regularized empirical risk minimization problems.
- Prox-SVRG [109]: we use the epoch length $m = 2n$ as they suggested. (Recall that Prox-SVRG is designed for strongly convex objectives and a dummy regularizer needs to be added in theory. However, in our experiments, we observed that this dummy regularizer is not necessary, so we have neglected it for a clean comparison.)
- SVRG++ [2]: we use the initial epoch length $m_0 = \frac{n}{4}$ as they suggested.
- Prox-SAG [85]: this is a proximal version of the SAG method. We note that the convergence of this Prox-SAG method has not been established in general. Nevertheless it demonstrates good performance in practice [109].
- SAGA [22]: this method requires an additional $n \times p$ matrix to store the gradient information.

In all the above algorithms except for Proximal-Newton and OWLQN, we tuned the step length carefully from the set $\{a \times 10^{-k} \mid a \in \{1, 5\}, k \in \mathbb{Z}\}$. All algorithms are initialized with a zero vector. For iSSPN, we implement both uniform and non-uniform sampling methods with the same sample size and try both *damped* and *full* step-size. In the subproblems, we set $\theta_t = 0.99$ as the inexactness parameter in (4.6). We check this criterion every 20 iterations in the FISTA. In Figure 4.1, (N)UD represents the damped-step iSSPN with (non-)uniform sampling theme while (N)UF is short for the full-step iSSPN with (non-)uniform sampling theme.

From our experiments, we observe the following facts:

- iSSPN with damped step-size converges faster than Proximal-Newton and OWLQN in all cases, indicating that they do improve over these second-order methods in the $n \gg p$ regime.

- The local convergence of iSSPN is comparable with these state-of-the-art first-order methods as we can see that iSSPN with full step-size consistently outperforms them in all datasets with different regularization terms.
- The non-uniform sampling scheme is more robust in terms of sample sizes needed to guarantee convergence and shows better convergence results compared with the uniform sampling scheme in all examples.

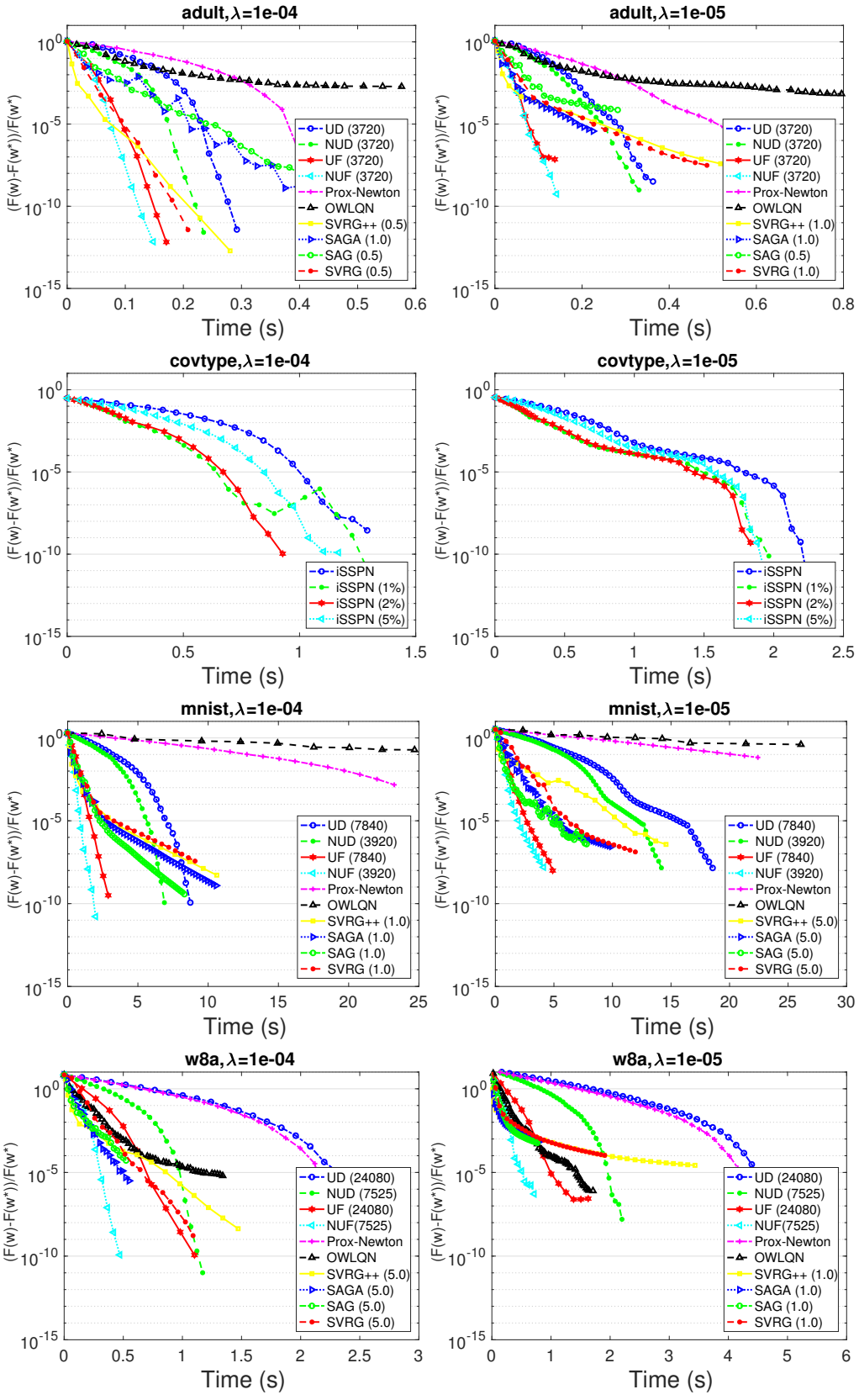


Figure 4.1: ℓ_1-regularized logistic regression: comparison of different methods on four datasets.

Effects of the inexactness: θ_t. Figure 4.2 shows the convergence of Algorithm 4.1 with different, constant values of the parameter θ_t when we solve problem (4.21) with $\lambda = 10^{-4}, 10^{-5}$ on the `MNIST` dataset. These figures confirm our conclusions about the effect of θ_t in the theoretical analysis.

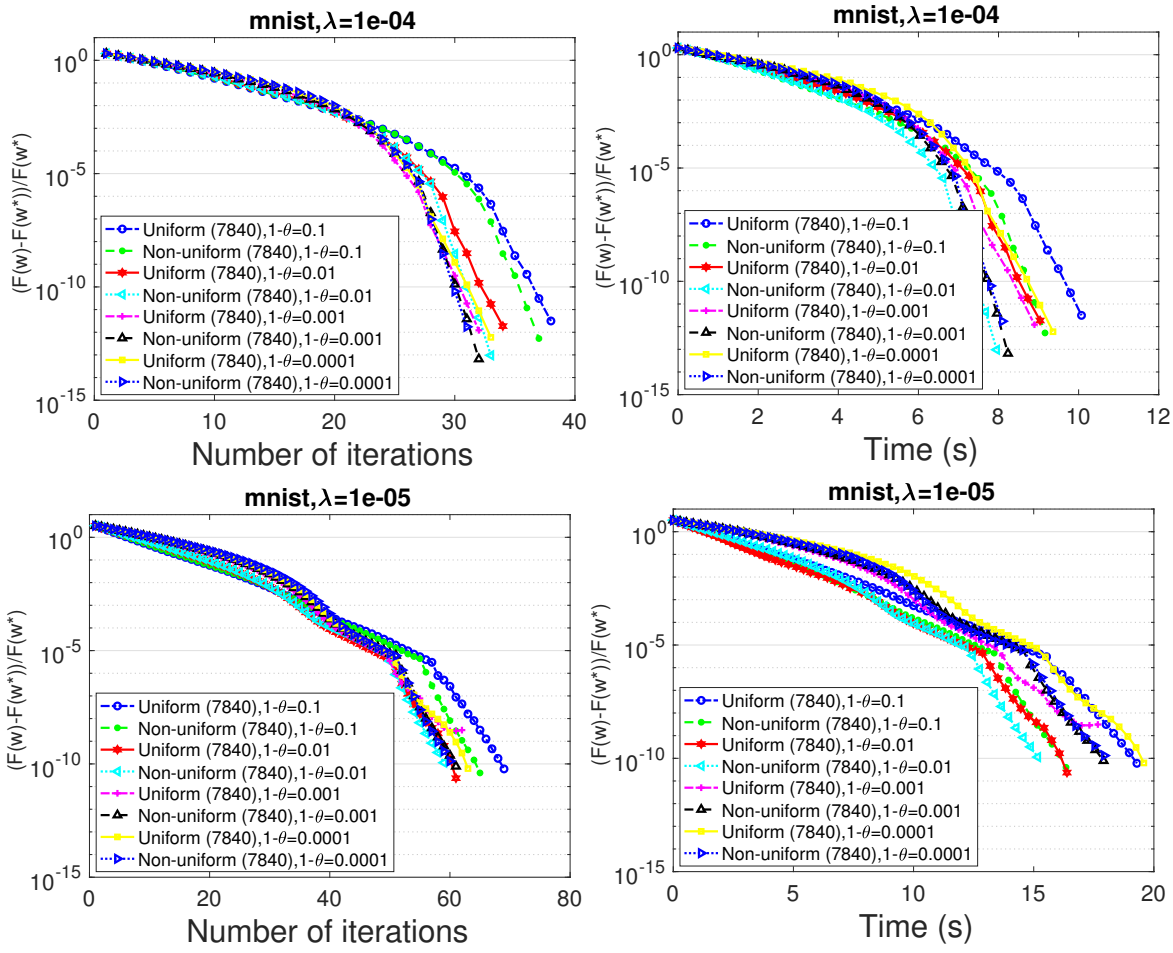


Figure 4.2: iSSPN on the `MNIST` dataset with different accuracy level θ_t in the subproblem (4.5).

It also shows that iSSPN can reach a high accuracy, even with very inaccurate solutions of the subproblems. In addition, the plots suggest there is an optimal value of θ_t that gives the fastest convergence.

Inexact gradient estimates. We further investigate the performance of iSSPN with subsampled gradient estimates on the `Covtype` dataset. Here, we fix the sample size 5500 for the Hessian estimate and choose a set of sample sizes $\{0.01n, 0.02n, 0.05n\}$ to approximate the gradient in the first few iterations of iSSPN. The performance is shown in Figure 4.3. In general, the inexact gradient estimate speeds up iSSPN even though more iterations are needed compared with iSSPN using exact gradient.

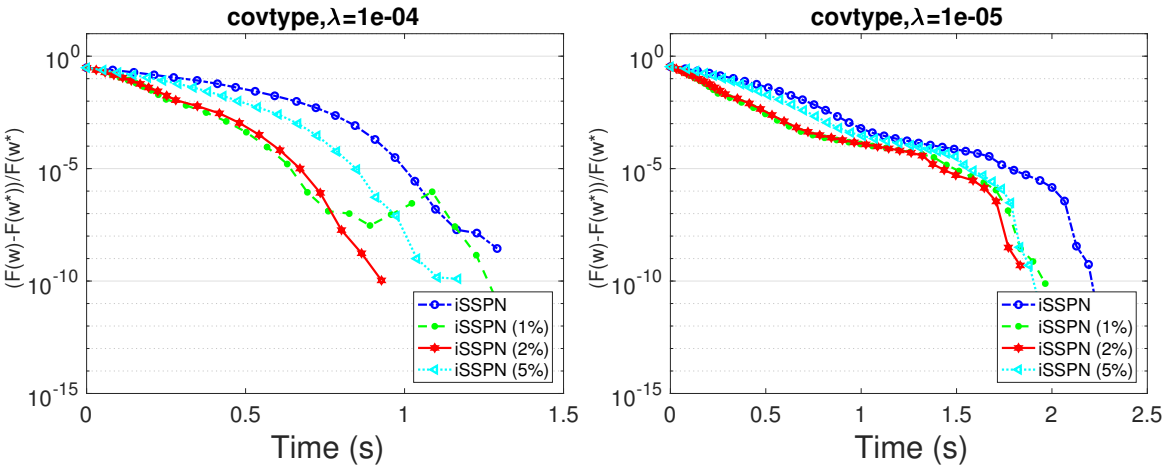


Figure 4.3: iSSPN on the `Covtype` dataset with inexact gradient estimate $\widetilde{\nabla} f(w_t)$.

The performance of iSSPN in high-dimension case. To test the effectiveness of iSSPN in high dimension settings, we use the `Real-Sim` dataset $(n, p) = (72,309, 20,958)$ and simulate two other datasets with $(n, p) = (50,000, 5,000)$ and $(100,000, 10,000)$. In all cases, we the variable $w \in \mathbb{R}^p$ is divided into $k = 20$ blocks sequentially and equally. At each iteration, we take two different ways to select the block: cyclic and uniformly random, and use the uniform sampling strategy to select $c = \min(0.05n, 20\frac{p}{k})$ samples from the datasets. We compare it with SVRG, SAGA with the best tuned step-size and the randomized block proximal damped Newton (RBPDN) proposed in [62] and the results are shown in Figure 4.4. Overall, the performance of iSSPN with block coordinate descent is competitive with SVRG and SAGA in both real and synthetic datasets.

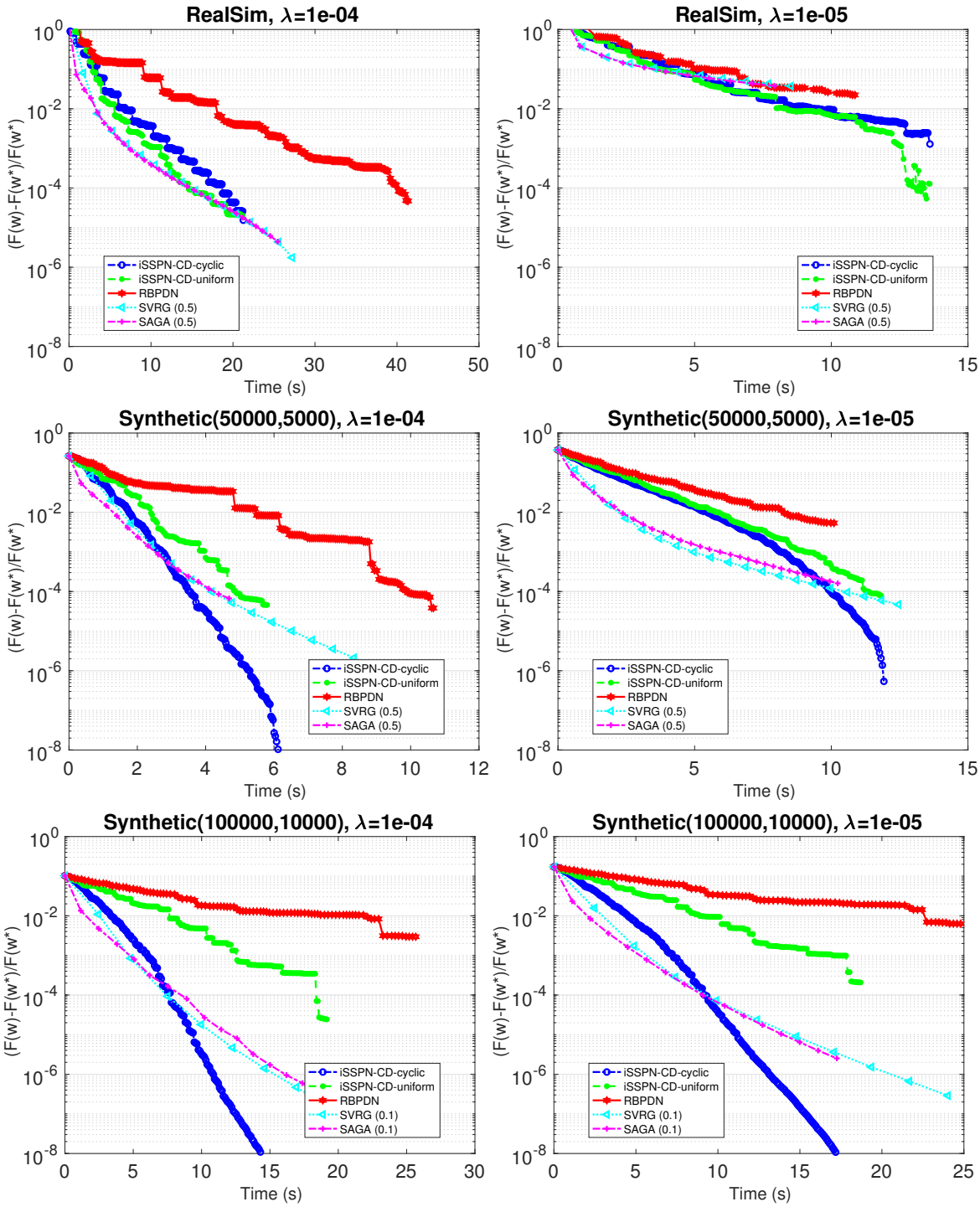


Figure 4.4: iSSPN with block coordinate descent in the high dimension setting.

4.4.2 Sparse Poisson Regression

We consider the sparse Poisson regression with penalized weighted score function proposed in [51] and use Algorithm 4.1 to solve the sparse Poisson regression problem in (4.3).

To perform the test, we generate matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ using standard sparse Gaussian distribution with 5% nonzero entries, *i.e.*, $\mathbf{X} = \text{sprandn}(n, p, 0.05)$. We set the number of nonzero elements of $w^\natural$ as $\lceil 0.2p \rceil$ and each of element randomly distributed from $\mathcal{N}(0, 1)$. Finally, we generate the data $y_i \sim \text{Poisson}(\exp(\mathbf{X}w^\natural))$ for $i = 1, \cdots, n$. For the penalty parameter, we choose $\lambda = (\sqrt{n})^{-1}\Psi^{-1}\big(1 - \frac{\alpha}{2p}\big)$ and take $\alpha = 0.05$ as normal cases. For the size of problem (4.3), we choose the following four different sets of (n, p): (50000,500), (200000,500), (200000,1000) and (500000,1000).

We compare iSSPN using the damped step-size with four first-order methods and the proximal-Newton method mentioned before measured by relative-error of the loss function vs. running time. We notice that the loss function $f \notin \mathcal{F}_L$ which indicates these first-order methods are not guaranteed to converge in theory. In the experiment, we tuned the step-size carefully from the set $\big\{a \times 10^{-k} \mid a \in \{1, 5\}, k \in \mathbb{Z}\big\}$.

From Figure 4.5, it can be seen that the damped-step iSSPN makes significant improvement over the Proximal-Newton method and is comparable with all other state-of-the-art first-order algorithms in all experiments. Using the theory of *generalized self-concordance*, we can choose a good step-size in the global region which guarantees sufficient decrease in function values and then switch from Phase 1 to Phase 2 automatically when the step-size η_t is sufficiently large (Figure 4.6).

The performance of iSSPN in high-dimension case. Here, we provide more numerical examples related to Sparse Poisson regression when p is large. We simulate two datasets with $(n, p) = (50,000, 10,000)$ and $(100,000, 10,000)$. In all cases, we the variable $w \in \mathbb{R}^p$ is divided into $k = 20$ blocks and we use the uniform sampling strategy to select $c = \min(0.05n, 20\frac{p}{k})$ samples from the datasets. We compare it with SVRG, SAGA with the best tuned step-size and RBPDN, and the results are shown as follows. Overall, the performance of iSSPN with block coordinate descent is better than RBPDN and competitive with SVRG and SAGA in practice.

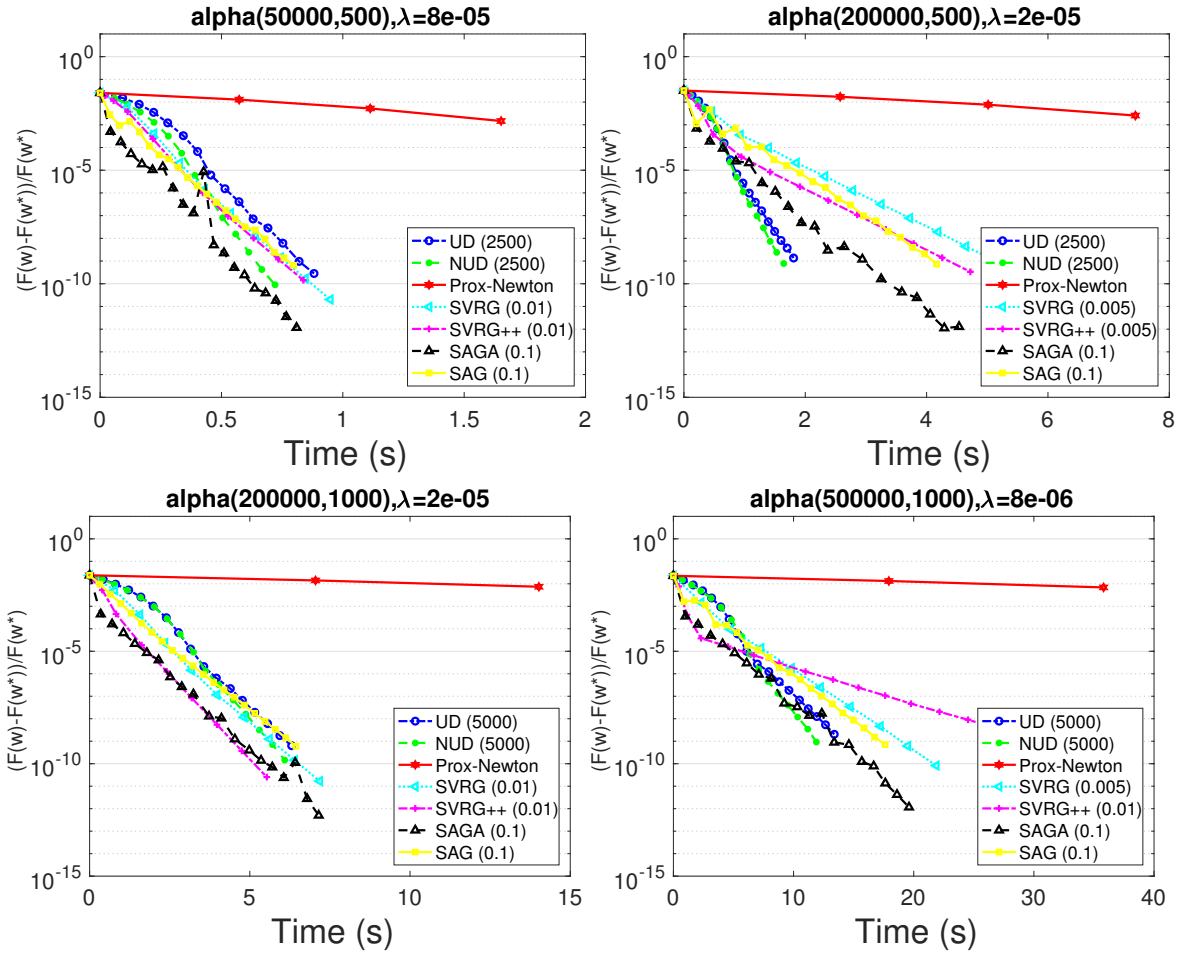


Figure 4.5: Sparse Poisson regression: comparison of different methods on synthetic datasets.

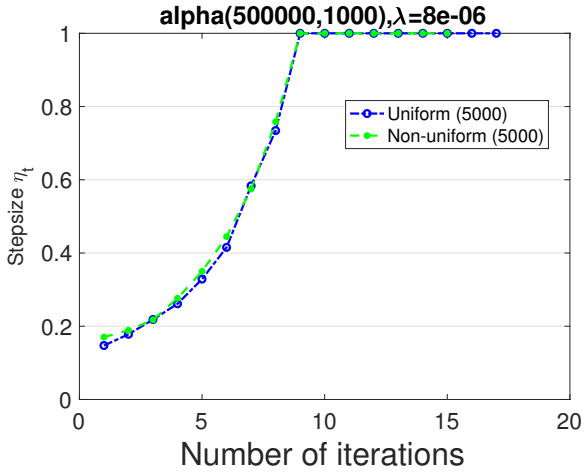
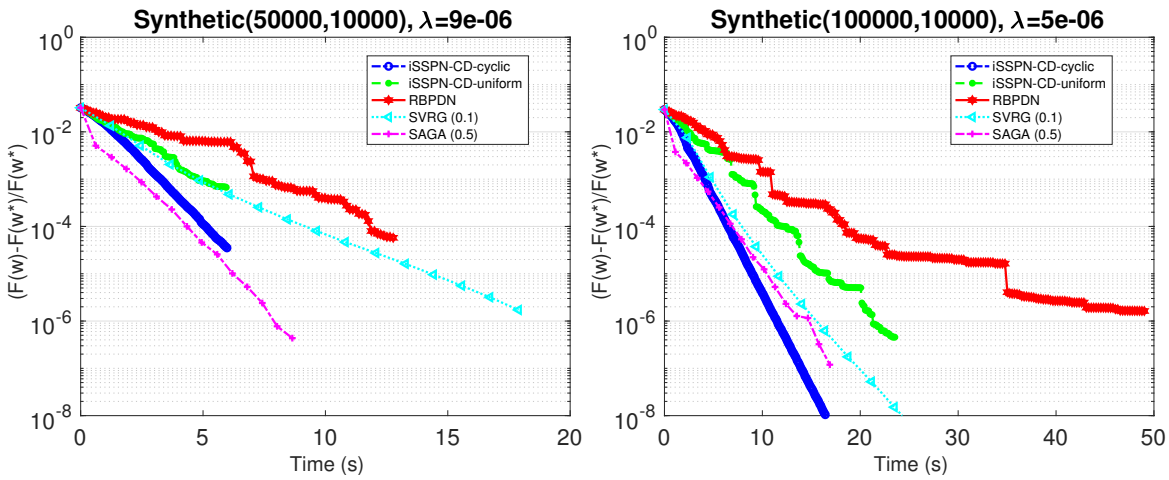


Figure 4.6: Sparse Poisson regression: a progress of step-size η_t over iterations.



4.5 Proofs of technical results

This supplementary document provides the full proof of our technical results presented in the main text. We also provide the details of how to approximately solve the subproblem (4.5) in practice.

4.5.1 Useful bounds for generalized self-concordant functions

Let f be a $(M_f, 2)$-*generalized self-concordant* function. For any $x, y \in \mathrm{dom}(f)$, we define the following quantities

$$d_2(x, y) := M_f \|y - x\|_2. \tag{4.22}$$

We first import some useful bounds for the class of $(M_f, 2)$-*generalized self-concordant* function, see [93].

Proposition 4.3. *For any $x, y \in \mathrm{dom}(f)$, let $d_\nu(x, y)$ be defined by (4.22) and define*

$$\bar{\omega}_2(\tau) := \frac{e^\tau - 1}{\tau} \text{ and } \omega_2(\tau) := \frac{e^\tau - \tau - 1}{\tau^2}.$$

Then, we have

B.1 $e^{-d_2(x,y)}\nabla^2 f(x) \preceq \nabla^2 f(y) \preceq e^{d_2(x,y)}\nabla^2 f(x)$.

B.2 $\bar{\omega}_2\left(-d_2(x,y)\right)\|y - x\|_x^2 \leq \langle \nabla f(y) - \nabla f(x), y - x\rangle \leq \bar{\omega}_2\left(d_2(x,y)\right)\|y - x\|_x^2$.

B.3 $\omega_2(-d_2(x,y))||y - x||_x^2 \leq f(y) - f(x) - \langle \nabla f(x), y - x\rangle \leq \omega_2(d_2(x,y))||y - x||_x^2$.

4.5.2 The proof of Lemmas 4.1 and 4.2

Given $\varphi_i : \mathbb{R} \to \mathbb{R}$ and $a_i \in \mathbb{R}^p$ and $b_i \in \mathbb{R}, i = 1, \cdots, n$, we consider the function $f : \mathbb{R}^p \to \mathbb{R}$ defined in the following form:

$$f(w) := \frac{1}{n} \sum_{i=1}^{n} \varphi_i(a_i^\top w + b_i). \tag{4.23}$$

Then the following propositions [93] holds:

Proposition 4.4. (a) *If φ_i in (4.23) are $(M_{\varphi_i}, 2)$-generalized self-concordant for $i = 1, \cdots, n$ and $M_{\varphi_i} \geq 0$, then f defined in (4.23) is also $(M_f, 2)$-generalized self-concordant with the constant $M_f := \max\{M_{\varphi_i} \|a_i\| \, | 1 \leq i \leq n\}$.*

(b) *If g and h are $(M_g, 2)$- and $(M_h, 2)$-generalized self-concordant , respectively. Then for any $\alpha, \beta > 0$, $f := \alpha g + \beta h$ is also $(M_f, 2)-$generalized self-concordant with $M_f := \max\{M_g, M_h\}$.*

Proof. [*The proof of Lemma 4.1 and 4.2*] Firstly, we notice that both $\log(1 + e^{-s})$ and e^s are $(1, 2)$-*generalized self-concordant* . The results in Lemma 4.1 and 4.2 can be derived by directly applying Proposition 4.4. □

4.5.3 The proof of Theorem 4.2

Before proving the above theorem, we first introduce the following operator and corresponding lemma which will be useful later. Given $\mathbf{H} \in \mathcal{S}_{++}^p$ and a proper, closed and convex function $g : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, we define

$$\mathcal{P}_{\mathbf{H}}^g(u) := (\mathbf{H} + \partial g)^{-1}(u) = \operatorname*{argmin}_w \Big\{ g(w) + \tfrac{1}{2}\langle \mathbf{H}w, w\rangle - \langle u, w\rangle \Big\}.$$

If $\mathbf{H} = \nabla^2 f(w)$ is the Hessian mapping of a strictly convex function f, then we can also write $\mathcal{P}_{\nabla^2 f(w)}(u)$ shortly as $\mathcal{P}_w(u)$ for our notational convenience. The following lemma will be used in the sequel, whose proof can be found in [99].

Lemma 4.4. *Let $g : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ be a proper, closed, and convex function, and $\mathbf{H} \in \mathcal{S}_{++}^p$. Then, the mapping $\mathcal{P}_{\mathbf{H}}^g$ defined above is non-expansive with respect to the weighted norm defined by*

$\mathbf{H}$, *i.e., for any* $u, v \in \mathbb{R}^p$, *we have*

$$\|\mathcal{P}^g_{\mathbf{H}}(u) - \mathcal{P}^g_{\mathbf{H}}(v)\|_{\mathbf{H}} \leq \|u - v\|^*_{\mathbf{H}}.$$

Let us define

$$S_w(u) := \nabla^2 f(w)u - \nabla f(u) \text{ and } e_w(u, v) := \left[\nabla^2 f(w) - \nabla^2 f(u)\right](v - u), \tag{4.24}$$

for any vectors $w, u \in \mathrm{dom} f$ and $v \in \mathbb{R}^p$.

Proof. [*The proof of Theorem 4.2*] At iteration t, there exist r_t and $s_t \in \partial g(w_t + v_t)$ such that

$$r_t = \nabla f(w_t) + \mathbf{H}_t v_t + s_t, \ \|r_t\|^*_{\mathbf{H}_t} \leq (1 - \theta_t) \|v_t\|_{\mathbf{H}_t}.$$

Define

$$r'_t := r_t + (\nabla^2 f(w_t) - \mathbf{H}_t)v_t = \nabla f(w_t) + \nabla^2 f(w_t)v_t + s_t,$$

and we can derive the following bound for $\|r'_t\|^*_{w_t}$:

$$\begin{aligned}
\left\|r'_t\right\|^*_{w_t} &\leq \|r_t\|^*_{w_t} + \left\|(\nabla^2 f(w_t) - \mathbf{H}_t)v_t\right\|^*_{w_t} \\
&\leq \sqrt{1 + \beta_t} \|r_t\|^*_{\mathbf{H}_t} + \beta_t \|v_t\|_{w_t} \\
&\leq \sqrt{1 + \beta_t}(1 - \theta_t) \|v_t\|_{\mathbf{H}_t} + \beta_t \|v_t\|_{w_t} \\
&\leq (1 + \beta_t)(1 - \theta_t) \|v_t\|_{w_t} + \beta_t \|v_t\|_{w_t} \\
&= \gamma_t \lambda_t.
\end{aligned} \tag{4.25}$$

Using Proposition 4.3, we obtain

$$f(w_{t+1}) \leq f(w_t) + \eta_t \nabla f(w_t)^\top v_t + \eta_t^2 \omega_2(\eta_t d_2(v_t))\lambda_t^2.$$

In view of convexity of g, we can derive

$$\begin{aligned} g(w_{t+1}) &\leq g(w_t) + \eta_t \left(g(w_t + v_t) - g(w_t)\right) \\ &\leq g(w_t) + \eta_t \left(r_t' - \nabla f(w_t) - \nabla^2 f(w_t) v_t\right)^\top v_t \\ &\leq g(w_t) + \eta_t \left(r_t' - \nabla f(w_t)\right)^\top v_t - \eta_t \lambda_t^2. \end{aligned}$$

Summing up the two last inequalities, we obtain the following estimate

$$\begin{aligned} F(w_{t+1}) &\leq F(w_t) + \eta_t \left(r_t' v_t - \lambda_t^2\right) + \eta_t^2 \omega_2(\eta_t d_2(v_t)) \lambda_t^2 \\ &\leq F(w_t) + \eta_t (||r_t'||_{w_t}^* \lambda_t - \lambda_t^2) + \eta_t^2 \omega_2(\eta_t d_2(v_t)) \lambda_t^2 \\ &\leq F(w_t) + \eta_t (\gamma_t \lambda_t^2 - \lambda_t^2) + \eta_t^2 \omega_2(\eta_t d_2(v_t)) \lambda_t^2 \\ &\leq F(w_t) - \Delta_t, \end{aligned}$$

where $\Delta_t := \left(\eta_t(\gamma_t - 1) + \eta_t^2 \omega_2(\eta_t d_2(v_t))\right) \lambda_t^2 > 0$ and the maximum attained at

$$\eta_t := \frac{\ln(1 + (1 - \gamma_t) d_2(v_t))}{d_2(v_t)}$$

after a few elementary calculations.

For the second part, we consider the distance between w_{t+1} and $w^\star$, e.g., $\|w_{t+1} - w^\star\|_{w^\star}$. By the definition of w_{t+1}, we have

$$\|w_{t+1} - w^\star\|_{w^\star} \leq (1 - \eta_t) \|w_t - w^\star\|_{w^\star} + \eta_t \|\bar{w}_t - w^\star\|_{w^\star} \tag{4.26}$$

Let $\bar{w}_t = w_t + v_t$ and use the notations in (4.24), then from the optimality condition of (4.1) and the inexact criterion in (4.6), we have $\bar{w}_t = \mathcal{P}_{w^\star}^g \left(S_{w^\star}(w_t) + e_{w^\star}(w_t, \bar{w}_t) + r_t\right)$ and $w^\star = \mathcal{P}_{w^\star}^g \left(S_{w^\star}(w^\star)\right)$. By Lemma 4.4 and the triangle inequality, one can show that

$$\|\bar{w}_t - w^\star\|_{w^\star} \leq \|S_{w^\star}(w_t) - S_{w^\star}(w^\star)\|_{w^\star}^* + \|e_{w^\star}(w_t, \bar{w}_t)\|_{w\star}^* + \|r_t\|_{w\star}^*. \tag{4.27}$$

From the same argument of the proof in [93, Theorem 5], we can derive

$$\|S_{w^\star}(w_t) - S_{w^\star}(w^\star)\|_{w^\star}^* \leq R(M_f \|w_t - w^\star\|_2)(M_f \|w_t - w^\star\|_2) \|w_t - w^\star\|_{w^\star} ,$$

where $R(t) := \left(\frac{3}{2} + \frac{t}{3}\right) e^t$. For notational simplicity, we denote $d_t^\star := M_f \|w_t - w^\star\|_2$. Then, the above inequality reduces to

$$\|S_{w^\star}(w_t) - S_{w^\star}(w^\star)\|_{w^\star}^* \leq R(d_t^\star) d_t^\star \|w_t - w^\star\|_{w^\star} . \tag{4.28}$$

Next, using the same proof in [93, Theorem 6], we can bound the second term $\|e_{w^\star}(w_t, \bar{w}_t)\|_{w\star}^*$ of (4.27) as

$$\|e_{w^\star}(w_t, \bar{w}_t)\|_{w\star}^* \leq (e^{d_t^\star} - 1) \|\bar{w}_t - w_t\|_{w^\star} . \tag{4.29}$$

For the last term in (4.27), with the same argument in (4.25), we have the following bound

$$\begin{aligned} \|r_t\|_{w\star}^* &\leq e^{0.5 d_t^\star} \|r_t\|_{w_t}^* \\ &\leq e^{0.5 d_t^\star} (1 - \theta_t)(1 + \beta_t) \|\bar{w}_t - w_t\|_{w_t} \\ &\leq e^{d_t^\star} (1 - \theta_t)(1 + \beta_t) \|\bar{w}_t - w_t\|_{w^\star} \end{aligned} \tag{4.30}$$

where the first and last inequalities result from the Hessian bound. Combining (4.27) with (4.28), (4.29), (4.30), we obtain

$$\|\bar{w}_t - w^\star\|_{w^\star} \leq (e^{d_t^\star}(\gamma_t + 1) - 1) \|\bar{w}_t - w_t\|_{w^\star} + R(d_t^\star) d_t^\star \|w_t - w^\star\|_{w^\star} .$$

With the triangle inequality $\|\bar{w}_t - w^\star\|_{w^\star} \geq \|\bar{w}_t - w_t\|_{w^\star} - \|w_t - w^\star\|_{w^\star}$, we can conclude

$$\|\bar{w}_t - w_t\|_{w^\star} \leq \underbrace{\frac{R(d_t^\star) d_t^\star + 1}{(2 - e^{d_t^\star}(\gamma_t + 1))}}_{\hat{R}_{t,1}(d_t^\star)} \|w_t - w^\star\|_{w^\star} \tag{4.31}$$

and

$$\|\bar{w}_t - w^\star\|_{w^\star} \leq \underbrace{\frac{R(d_t^\star) d_t^\star + (e^{d_t^\star}(\gamma_t + 1) - 1)}{(2 - e^{d_t^\star}(\gamma_t + 1))}}_{\hat{R}_{t,2}(d_t^\star)} \|w_t - w^\star\|_{w^\star} . \tag{4.32}$$

Using this above bound, (4.26) and the fact $\eta_t \leq 1$, we can bound

$$\|w_{t+1} - w^\star\|_{w^\star} \leq [(1-\eta_t) + \hat{R}_{t,2}(d_t^\star)] \|w_t - w^\star\|_{w^\star}. \tag{4.33}$$

With the step-size $\eta_t = \frac{\ln(1+(1-\gamma_t)d_2(v_t))}{d_2(v_t)}$, we can bound $1-\eta_t$ as

$$\begin{aligned}
1-\eta_t &= 1 - \frac{\ln(1+(1-\gamma_t)d_2(v_t))}{d_2(v_t)} \leq 1-(1-\gamma_t)\left(1-(1-\gamma_t)\frac{d_2(v_t)}{2}\right) \\
&= \gamma_t + \frac{(1-\gamma_t)^2 M_f \|\bar{w}_t - w_t\|_2}{2} \\
&\leq \gamma_t + M_f \frac{(1-\gamma_t)^2 \|\bar{w}_t - w_t\|_{w^\star}}{2\sqrt{\underline{\sigma}^\star}} \\
&\overset{(4.31)}{\leq} \gamma_t + M_f \frac{(1-\gamma_t)^2 \hat{R}_{t,1}(d_t^\star)}{2\sqrt{\underline{\sigma}^\star}} \|w_t - w^\star\|_{w^\star}
\end{aligned}$$

Finally, we get

$$\|w_{t+1} - w^\star\|_{w^\star} \leq (\gamma_t + M_f \frac{(1-\gamma_t)^2 \hat{R}_{t,1}(d_t^\star)}{2\sqrt{\underline{\sigma}^\star}}) \|w_t - w^\star\|_{w^\star}^2 + \hat{R}_{t,2}(d_t^\star) \|w_t - w^\star\|_{w^\star}. \tag{4.34}$$

Notice that both $\hat{R}_{t,1}(d_t^\star)$ and $\hat{R}_{t,2}(d_t^\star)$ are increasing function with respect to $d_t^\star$ and $\hat{R}_{t,1}(0) = \frac{1}{1-\gamma_t}$, $\hat{R}_{t,2}(0) = \frac{\gamma_t}{1-\gamma_t}$. With $\sup_t\{\gamma_t\} \leq 0.2$, there exists a constant $\bar{d}$ such that if $0 \leq d_t^\star \leq \bar{d} \approx 0.05$, then $\hat{R}_{t,1} \leq 2$ and $\hat{R}_{t,2} \leq 0.5$. The last estimate shows that if $\|w_0 - w^\star\|_{w^\star} \leq \min\{\frac{1}{2(\sup_t\{\gamma_t\}+M_f/\sqrt{\underline{\sigma}^\star})}, \frac{\bar{d}\sqrt{\underline{\sigma}^\star}}{M_f}\}$, then $\{\|w_t - w^\star\|_{w^\star}\}$ converges to zero in a linear-quadratic rate. On the other hand, by the definition of $\underline{\sigma}^\star$, we have $\sqrt{\underline{\sigma}^\star}\|w_t - w^\star\|_2 \leq \|w_t - w^\star\|_{w^\star}$ which implies that $\{\|w_t - w^\star\|_2\}$ also converges to zero in a linear-quadratic rate. □

4.5.4 The proof of Theorem 4.3: The full-step variant of Algorithm 4.1

At iteration $t+1$, we define $s_{t+1} := r_{t+1} - \nabla f(w_{t+1}) - \mathbf{H}_{t+1}v_{t+1} \in \partial g(w_{t+1}+v_{t+1})$. By the monotonicity of the subdifferential ∂g, we have

$$(s_{t+1}-s_t)^\top v_{t+1} = (s_{t+1}-s_t)^\top (w_{t+1}+v_{t+1}-w_t-v_t) \geq 0.$$

This observation applies to the first inequality of the following derivation:

$$\begin{aligned}
\|v_{t+1}\|_{w_{t+1}} &\leq \|v_{t+1} + \nabla^2 f(w_{t+1})^{-1}(s_{t+1} - s_t)\|_{w_{t+1}} \\
&= \|\nabla^2 f(w_{t+1})^{-1}(\nabla^2 f(w_{t+1})v_{t+1} + s_{t+1} - s_t)\|_{w_{t+1}} \\
&= \|r'_{t+1} - r'_t + \nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|^*_{w_{t+1}} \\
&\leq \|r'_{t+1}\|^*_{w_{t+1}} + \|r'_t\|^*_{w_{t+1}} + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|^*_{w_{t+1}} \\
&\leq \gamma_{t+1}\|v_{t+1}\|_{w_{t+1}} + \|r'_t\|^*_{w_{t+1}} + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|^*_{w_{t+1}}.
\end{aligned}$$

Rearranging the above inequality, we have

$$\begin{aligned}
(1-\gamma_{t+1})\|v_{t+1}\|_{w_{t+1}} &\leq \|r'_t\|^*_{w_{t+1}} + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|^*_{w_{t+1}} \\
&\leq e^{0.5d_2(v_t)}\left(\|r'_t\|^*_{w_t} + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|^*_{w_t}\right) \\
&\leq e^{0.5d_2(v_t)}\left(\gamma_t + \frac{e^{d_2(v_t)} - 1}{d_2(v_t)} - 1\right)\|v_t\|_{w_t},
\end{aligned}$$

where the last inequality results from [93, Lemma 2]. Notice that $\|v_t\|_{w_t} \geq \frac{\sqrt{\underline{\sigma}_t} d_2(v_t)}{M_f}$ and $\underline{\sigma}_{t+1}^{-1} \leq e^{d_2(v_t)}\underline{\sigma}_t^{-1}$. It follows from the above inequality that

$$\begin{aligned}
(1-\gamma_{t+1})\frac{\|v_{t+1}\|_{w_{t+1}}}{\sqrt{\underline{\sigma}_{t+1}}} &\leq e^{d_2(v_t)}\left(\gamma_t + \frac{e^{d_2(v_t)} - 1}{d_2(v_t)} - 1\right)\frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}} \\
&\leq e^{d_2(v_t)}\left(\gamma_t + d_2(v_t)\right)\frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}} \\
&\leq e^{d_2(v_t)}\left(\gamma_t + M_f\frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}}\right)\frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}}
\end{aligned}$$

holds when $d_2(v_t) \leq 1$. If $\frac{\|v_0\|_{w_0}}{\sqrt{\underline{\sigma}_0}} \leq \frac{1}{4M_f}$, one can show the following:

$$\frac{\|v_{t+1}\|_{w_{t+1}}}{\sqrt{\underline{\sigma}_{t+1}}} \leq \frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}}$$

as $d_2(v_0) \leq M_f\frac{\|v_0\|_{w_0}}{\sqrt{\underline{\sigma}_0}} \leq 0.25$ and $\gamma_0 \leq 0.2$ hold. The above inequality shows that the ratio $\left\{\frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}}\right\}$ converges to zero at a linear-quadratic rate. Consequently, $d_2(v_t)$ also converges linear-quadratically

to zero since $d_2(v_t) \leq M_f \frac{\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}}$. For the second part, based on $B.2$ in Proposition 4.3, we have

$$\bar{\omega}(-d_t^\star) \|w_t - w^\star\|_{w_t}^2 \leq \langle \nabla f(w_t) - \nabla f(w^\star), w_t - w^\star \rangle.$$

By the optimality condition of (4.1), we have

$$s^\star + \nabla f(w^\star) = 0$$

for some $s^\star \in \partial g(w^\star)$. Combining this with $\nabla f(w_t) = r_t - \mathbf{H}_t v_t - s_t$ for some $s_t \in \partial g(w_t + v_t)$, we obtain

$$\begin{aligned}
\langle \nabla f(w_t) - \nabla f(w^\star), w_t - w^\star \rangle &= \langle r_t - \mathbf{H}_t v_t, w_t - w^\star \rangle - \underbrace{\langle s_t - s^\star, w_t - w^\star \rangle}_{\geq 0} \\
&\leq \langle r_t' - \nabla^2 f(w_t) v_t, w_t - w^\star \rangle \\
&\leq \left(\|r_t'\|_{w_t}^\star + \left\|\nabla^2 f(w_t) v_t\right\|_{w_t}^* \right) \|w_t - w^\star\|_{w_t} \\
&\leq (\gamma_t + 1) \|v_t\|_{w_t} \|w_t - w^\star\|_{w_t}.
\end{aligned}$$

Hence, we obtain

$$\bar{\omega}(-d_t^\star) \|w_t - w^\star\|_{w_t} \leq (\gamma_t + 1) \|v_t\|_{w_t}.$$

By the definition of $\bar{\omega}$, one can show that $\bar{\omega}(-d_t^\star) = \frac{1 - e^{-d_t^\star}}{d_t^\star} \geq 1 - \frac{d_t^\star}{2} \geq 1 - \frac{1}{2}$, whenever $d_t^\star \leq 1$. Using above inequality, we have $\|w_t - w^\star\|_{w_t} \leq 2(\gamma_t + 1) \|v_t\|_{w_t}$. On the other hand, by the definition of $\underline{\sigma}_t$, we have $\sqrt{\underline{\sigma}_t} \|w_t - w^\star\|_2 \leq \|w_t - w^\star\|_{w_t}$. With the last two inequalities, we obtain $\|w_t - w^\star\|_2 \leq \frac{2(\gamma_t+1)\|v_t\|_{w_t}}{\sqrt{\underline{\sigma}_t}} = \frac{2(\gamma_t+1)\lambda_t}{\sqrt{\underline{\sigma}_t}}$ provided $d_t^\star \leq 1$. Since $\left\{ \frac{\lambda_t}{\sqrt{\underline{\sigma}_t}} \right\}$ linear-quadratically converges to zero, the last relation also shows that $\{\|w_t - w^\star\|_2\}$ converges to zero in a linear-quadratic rate. □

4.5.5 The proof of Theorem 4.5: Convergence of the second variant

Similar to the proof in Appendix 4.5.3, at iteration t, there exist $\tilde{r}_t$ and $s_t \in \partial g(w_t + v_t)$ such that

$$\tilde{r}_t = \widetilde{\nabla} f(w_t) + \mathbf{H}_t v_t + s_t, \ \|\tilde{r}_t\|_{\mathbf{H}_t}^* \leq (1 - \theta_t) \|v_t\|_{\mathbf{H}_t}.$$

Define

$$\tilde{r}_t' := \tilde{r}_t + (\nabla^2 f(w_t) - \mathbf{H}_t)v_t + (\nabla f(w_t) - \widetilde{\nabla} f(w_t)) = \nabla f(w_t) + \nabla^2 f(w_t)v_t + s_t,$$

and we can derive the following bound for $\|\tilde{r}_t'\|_{w_t}^*$:

$$\begin{aligned} \left\|\tilde{r}_t'\right\|_{w_t}^* &\leq \|\tilde{r}_t\|_{w_t}^* + \left\|(\nabla^2 f(w_t) - \mathbf{H}_t)v_t\right\|_{w_t}^* + \|\nabla f(w_t) - \widetilde{\nabla} f(w_t)\|_{w_t}^* \\ &\leq \underbrace{(\gamma_t + \xi_t)}_{\tilde{\gamma}_t} \lambda_t, \end{aligned} \tag{4.35}$$

where the second inequality follows from the same observation in (4.25) and condition (C2). By replacing r_t', γ_t with $\tilde{r}_t', \tilde{\gamma}_t$ respectively in the first part of Appendix 4.5.3, we obtain the following estimate

$$\begin{aligned} F(w_{t+1}) &\leq F(w_t) + \eta_t \left(\tilde{r}_t' v_t - \lambda_t^2\right) + \eta_t^2 \omega_2(\eta_t d_2(v_t))\lambda_t^2 \\ &\leq F(w_t) + \eta_t(||\tilde{r}_t'||_{w_t}^* \lambda_t - \lambda_t^2) + \eta_t^2 \omega_2(\eta_t d_2(v_t))\lambda_t^2 \\ &\leq F(w_t) + \eta_t(\tilde{\gamma}_t \lambda_t^2 - \lambda_t^2) + \eta_t^2 \omega_2(\eta_t d_2(v_t))\lambda_t^2 \\ &\leq F(w_t) - \tilde{\Delta}_t, \end{aligned}$$

where $\tilde{\Delta}_t := \left(\eta_t(1 - \tilde{\gamma}_t) - \eta_t^2 \omega_2(\eta_t d_2(v_t))\right) \lambda_t^2 > 0$ and the maximum attained at

$$\eta_t := \frac{\ln(1 + (1 - \tilde{\gamma}_t)d_2(v_t))}{d_2(v_t)}$$

after a few elementary calculations.

Following the same idea, the local linear-quadratic convergence rate of the damped-step iSSPN can be obtained by the same argument in the second part of Appendix 4.5.3 with r_t', γ_t replaced by $\tilde{r}_t', \tilde{\gamma}_t$.

For the last part, by the same argument in Appendix 4.5.4, one can show that

$$\lambda_{t+1} \leq \tilde{\gamma}_{t+1}\lambda_{t+1} + \|\tilde{r}_t'\|_{w_{t+1}}^* + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|_{w_{t+1}}^*$$

Rearranging the above inequality, under condition (C2), we have

$$\begin{aligned}(1-\tilde{\gamma}_{t+1})\lambda_{t+1} &\leq \|\tilde{r}_t'\|_{w_{t+1}}^* + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|_{w_{t+1}}^* \\ &\leq e^{0.5d_2(v_t)}\left(\|\tilde{r}_t'\|_{w_t}^* + \|\nabla f(w_{t+1}) - \nabla f(w_t) - \nabla^2 f(w_t)v_t\|_{w_t}^*\right) \\ &\leq e^{0.5d_2(v_t)}\left(\tilde{\gamma}_t + \frac{e^{d_2(v_t)}-1}{d_2(v_t)} - 1\right)\lambda_t.\end{aligned}$$

The rest of the proof follows the same argument as in the proof of Theorem 4.3 and we omit the details here. □

4.5.6 The proof of Proposition 4.2: Bounds on subsampled gradient

Before proving Proposition 4.2, we first import the following basic result from [84].

Lemma 4.5. *For a given $w \in \mathrm{dom}(F)$, let $\|\nabla f_i(w)\| \leq G(w)$, $i = 1, \cdots, n$. For any $0 < \epsilon < 1$ and $0 < \delta < 1$, if $|\mathcal{S}| \geq \frac{G(w)^2}{\epsilon^2}\left(1 + \sqrt{8\ln\frac{1}{\delta}}\right)^2$, then for $\widetilde{\nabla} f(w)$ defined in (4.12), we have*

$$\Pr\left(\|\nabla f(w) - \widetilde{\nabla} f(w)\| \leq \epsilon\right) \geq 1 - \delta.$$

Proof. [*The proof of Proposition 4.2*] Since $\|w_t - w^\star\| \leq \frac{1}{2M_f}$, from Proposition 4.3(B.1), the smallest eigenvalue of $\nabla^2 f(w_t)$ defined by $\sigma_{\min}(w_t) = \lambda_{\min}(\nabla^2 f(w_t))$ satisfying the following bound:

$$\sigma_{\min}(w_t) \geq e^{-0.25}\sigma_{\min}(w^\star).$$

By the definition of local dual norm at w_t, we obtain

$$\begin{aligned}\|\nabla f(w_t) - \widetilde{\nabla} f(w_t)\|_{w_t}^* &\leq \frac{1}{\sqrt{\sigma_{\min}(w_t)}}\|\nabla f(w_t) - \widetilde{\nabla} f(w_t)\| \\ &\leq \frac{e^{0.125}}{\sqrt{\sigma_{\min}(w^\star)}}\|\nabla f(w_t) - \widetilde{\nabla} f(w_t)\|.\end{aligned} \tag{4.36}$$

From the above inequality, we have the following bound

$$\begin{aligned}\Pr\left(\|\nabla f(w_t)-\widetilde{\nabla} f(w_t)\|_{w_t}^{*}\leq \xi_t\lambda_t\right) &\geq \Pr\left(\frac{e^{0.125}}{\sqrt{\sigma_{\min}(w^\star)}}\|\nabla f(w_t)-\widetilde{\nabla} f(w_t)\|\leq \xi_t\lambda_t\right)\\ &= \Pr\left(\|\nabla f(w_t)-\widetilde{\nabla} f(w_t)\|\leq e^{-0.125}\sqrt{\sigma_{\min}(w^\star)}\xi_t\lambda_t\right).\end{aligned}$$

Therefore, Proposition 4.2 follows from the above estimate and Lemma 4.5. □

4.5.7 The proof of Theorem 4.4: Sufficient sampling size

Here, we omit the uniform sampling and present the detailed proof of the non-uniform sampling. For notational simplicity, we denote $\mathbf{A}_i = \mathbf{A}_i(w_t), \mathbf{A} = \mathbf{A}(w_t)$ and $\mathbf{H} = \mathbf{H}_t$. For $i = 1, \cdots, n$, define

$$\mathbf{X}_i = \begin{cases} \left(\frac{1}{q_i}-1\right)\mathbf{A}_i\mathbf{A}_i^\top, & \text{with probability } q_i,\\ -\mathbf{A}_i\mathbf{A}_i^\top, & \text{with probability } 1-q_i,\end{cases} \tag{4.37}$$

and let $\mathbf{Y} = \sum_{i=1}^n \mathbf{X}_i = \mathbf{H} - \mathbf{A}\mathbf{A}^\top$. We have $\mathbb{E}[\mathbf{X}_i] = 0$. Let $\mathcal{I} = \{i \mid q_i = 1\}$, then if $i \in \mathcal{I}$, then $\|\mathbf{X}_i\| = 0$. If $i \notin \mathcal{I}$,

$$\|\mathbf{X}_i\| \leq \frac{\|\mathbf{A}_i\mathbf{A}_i^\top\|_2}{q_i} = \frac{\|\mathbf{A}_i\|_2^2}{c\cdot p_i} \leq \frac{1}{c}\underbrace{\max_{1\leq i\leq n}\left\{\frac{\|\mathbf{A}_i\|^2}{p_i}\right\}}_{\hat{\rho}_1}. \tag{4.38}$$

Next, we bound $\mathbb{E}[\mathbf{Y}^2] = \sum_{i=1}^n \mathbb{E}[\mathbf{X}_i^2]$. We have

$$\begin{aligned}\sum_{i=1}^n \mathbb{E}[\mathbf{X}_i^2] &= \sum_{i=1}^n (\frac{1}{q_i}-1)\mathbf{A}_i\mathbf{A}_i^\top\mathbf{A}_i\mathbf{A}_i^\top\\ &= \sum_{i\in\mathcal{I}}\left(\frac{1}{q_i}-1\right)\mathbf{A}_i\mathbf{A}_i^\top\mathbf{A}_i\mathbf{A}_i^\top\\ &\preceq \sum_{i\in\mathcal{I}}\frac{1}{c\cdot p_i}\mathbf{A}_i\mathbf{A}_i^\top\mathbf{A}_i\mathbf{A}_i^\top\\ &\preceq \frac{1}{c}\max_{1\leq i\leq n}\left\{\frac{\|\mathbf{A}_i\|^2}{p_i}\right\}\sum_{i\in\mathcal{I}}\mathbf{A}_i\mathbf{A}_i^\top\\ &\preceq \frac{\hat{\rho}_1}{c}\mathbf{A}\mathbf{A}^\top.\end{aligned} \tag{4.39}$$

Therefore, we have

$$\left\|\mathbb{E}[\mathbf{Y}^2]\right\| \leq \frac{\hat{\rho}_1}{c}\|\mathbf{A}\|_2^2. \tag{4.40}$$

Given (4.38), (4.40), and Theorem 1.4 in [100], $\|\sum_{i=1}^{n} \mathbf{X}_i\| \leq \epsilon$ holds with probability at least $1-\delta$, where

$$\delta = p \exp\left(\frac{-c\epsilon^2}{2\hat{\rho}_1(\|\mathbf{A}\|_2^2 + \epsilon/3)}\right).$$

Solving for ϵ gives

$$\epsilon = \tau_1 \hat{\rho}_1 + \sqrt{\tau_1 \hat{\rho}_1 \ \left(6\|\mathbf{A}\|_2^2 + \tau_1 \hat{\rho}_1\right)}, \qquad \tau_1 \equiv \frac{\ln(d/\delta)}{3c}.$$

With $p_i = \frac{r_i}{\sum_{j=1}^{n} r_j}$, we obtain the bound $\hat{\rho}_1 \leq \|\mathbf{A}\|_2^2 \tau_2$, where $\tau_2 \equiv \mathsf{sr}(\mathbf{A}) \geq 1$, and

$$\epsilon \leq \|\mathbf{A}\|_2^2 \left(\tau_1\tau_2 + \sqrt{\tau_1\tau_2 \ (6 + \tau_1\tau_2)}\right).$$

Let $\gamma_0 = \tau_1\tau_2$ and divide by $\|\mathbf{A}\|_2^2$, then we can conclude that for any $\delta > 0$, with probability at least $1-\delta$,

$$\frac{\left\|\mathbf{H} - \mathbf{A}\mathbf{A}^\top\right\|_2}{\|\mathbf{A}\mathbf{A}^\top\|_2} \leq \gamma_0 + \sqrt{\gamma_0 \ (6 + \gamma_0)}, \text{ where } \gamma_0 \equiv \mathsf{sr}(\mathbf{A}) \ \frac{\ln(d/\delta)}{3\,c}.$$

This completes the proof. □

CHAPTER 5: Hybrid Bayesian Optimization with DIRECT

5.1 Introduction

In this chapter, we aim to work on the *black-box* optimization with application in hyper-parameter optimization (HPO) for machine learning models. Many HPO problems come in the form of the following:

$$\max_{x \in \Omega \subset \mathbb{R}^p} f(x) \tag{5.1}$$

The feasible set and objective function typically have the following properties:

- The input $x \in \mathbb{R}^p$ is a low dimensional vector, e.g., $p \leq 10$.
- f is expensive to evaluate and often, the number of evaluations is typically limited to a few hundred. Such limitation arises when the function evaluation takes a significant amount of time (in hours) and bears a budget limitation (e.g., from purchasing cloud computing power).
- f lacks the special structure like concavity or linearity which makes many efficient algorithms unavailable. We refer such function f as black-box.
- Only function value of f can be accessed while no first-order or second-order information can be used. This is often called derivative-free optimization.
- We aim to find a **global** rather than the local optimum.

Black-box optimization has a long history and can be traced back to the deterministic direct-search (DDS) method proposed in [49]. Subsequently, many variants of DDS have been proposed, including the generalized pattern-search (GPS) method [97, 98], DIRECT [53] and the mesh adaptive direct search (MADS) method [4]. In addition, evolution strategy [86] is a class of black-box

optimization that are heuristic search procedures inspired by natural evolution, including the differential evolution (DE) [77] and the covariance matrix adaptation evolution strategy (CMA-ES) [38]. Another important class of black-box optimization is the local model-based methods in which the updates are based primarily on the predictions of a model that serves as a surrogate of the objective function or of a related merit function. For instance, RBFOpt [20] utilizes the radial basis function as the surrogate model.

However, most of the methods above can require extensive function evaluations which makes them not applicable in many situations as f is expensive to evaluate and often, the number of evaluations is typically limited to a few hundred. `DIRECT` [53] and Bayesian optimization [31, 68] are two powerful black-box optimization algorithms for global optimization. Each of them uses a different assumption on the unknown function, and they exploit different techniques to search the feasible domain. `DIRECT` is based on a space-partitioning strategy which is designed to adaptively perform local and global exploration at each iteration under the Lipschitz-continuity condition. A compelling benefit of `DIRECT` is the ability to quickly locate regions that potentially contain a global optimum [60, 59, 25]. However, the performance of `DIRECT` can deteriorate when the assumption is loosely satisfied. A Bayesian optimization method frequently assumes a smooth Gaussian process prior on the objective function. A closed-form posterior distribution is computed using the observed function evaluations. The algorithm uses an acquisition function to choose the next point to evaluate the function value. When the feasible domain is large, the BO assumption can be absent. As a result, its convergence can be slow. We propose a framework to combine two algorithms in order to take advantage of their strength.

5.2 Related Work

There are two main steps in a `DIRECT`-type algorithm: 1) selecting a sub-region within Ω, the so-called potentially optimal hyper-rectangle, in order to get a new sample point over the sub-region, and 2) splitting the potentially optimal hyper-rectangle. In the literature, a number of variants of `DIRECT` algorithms have been proposed to improve the performance of `DIRECT`, most of them are devoted to the first step [32, 30, 61, 50, 96, 57]. There are a limited number of papers working on the second step, for example [33]. Another line of research direction to speed up `DIRECT` is to

incorporate a local search strategy in the framework [60, 59, 25]. These local searches are mainly based on a surrogate model for the objective function, and do not use any prior knowledge of the function.

Recently, Bayesian Optimization, especially by making use of a Gaussian process (GP) generated using a Gaussian or Matérn kernel, is widely used in hyper-parameter tuning for machine learning models [31, 68]. The classical implementation of BO methods for some acquisition functions includes GP-UCB [91], and GP-EI [102]. Knowledge gradient [107] and max-value entropy search [105] are recently proposed to speed up the practical performance of BO.

We summarize our contributions as follows:

- We propose a sample-efficient hybrid Bayesian Optimization algorithm with DIRECT to solve (5.1) which combines strengths from `BO` and `DIRECT` .
- The global convergence of the new algorithm is developed.
- We demonstrate the practical performance of the new algorithm by solving synthetic functions and benchmark hyper-parameter tuning problems in machine learning.

5.3 Background

In this section, we review basic steps for designing the original `DIRECT` and Bayesian optimization methods. We can leverage them to show how we modify these steps to derive a hybrid algorithm.

5.3.1 DIRECT Algorithm

We consider the well-known `DIRECT` algorithm [53] for solving global optimization problems with box constraint. The name `DIRECT` comes from the shortening of the phrase "DIviding RECTangles, which describes the way the algorithm partitions the feasible domain by a number of hyper-rectangles in order to move towards the optimum. In the literature, a number of variants of `DIRECT` algorithms have been proposed for a general objective function [30, 50, 57, 25, 59] .

The `DIRECT` algorithm begins the optimization by transforming the domain of the problem (5.1)

linearly into the unit hyper-cube. Therefore, we assume for the rest of the paper that

$$\Omega = \{\mathbf{x} \in \mathbb{R}^p : 0 \leq x_i \leq 1\}. \tag{5.2}$$

In each iteration, Algorithm 5.1 consists of three main steps. First, we identify a set of *potentially optimal* hyper-rectangles based on a criterion. We expect that the sub-regions have a high chance to contain a global optimal solution. The second step is perform a local search over the potentially optimal hyper-rectangles. Thirdly, we divide the selected hyper-rectangles into smaller hyper-rectangles.

At the k-th iteration, let $\mathcal{P}_k$ define the set of the current hyper-rectangles associated with the index set $\mathcal{I}_k$

$$\mathcal{P}_k = \{\mathcal{H}_i : i \in \mathcal{I}_k\}$$

where

$$\mathcal{H}_i = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{0} \leq \mathbf{l}_i \leq \mathbf{x} \leq \mathbf{u}_i \leq \mathbf{1}\}$$

is a hyper-rectangle in the partition. The set $\mathcal{C}_k$ denotes the set of centers $\mathbf{c}_k$ of hyper-rectangles in $\mathcal{P}_k$. Denote f_i by the function value evaluated at the centre of $\mathcal{H}_i$ by evaluating at the current sampled points in the sub-region including its center $\mathbf{c}_i$. We use m to count the number of function evaluations and f_{eval}^{max} is the maximal number of function evaluations. We present the our main algorithm `DIRECT` in Algorithm 5.1.

Algorithm 5.1 `DIRECT`

Define $\mathbf{c}_1 = (0.5, \ldots, 0.5) \in \mathbb{R}^p$ and set $\mathbf{x}_{max} = \mathbf{c}_1, f_{max} = f(\mathbf{c}_1), k = m = 1$
Run the Initialization Step to get $\mathcal{P}_1, \mathcal{I}_1, \mathcal{C}_1, f_i(\forall i \in \mathcal{P}_1), f_{max}$ and $\mathbf{x}_{max}$
while $m \leq f_{eval}^{max}$ **do**
  Identify the set $\mathcal{S}$ of all potentially optimal hyper-rectangles in $\mathcal{P}_k$
  for $j \in \mathcal{S}$ **do**
    a) (Optional) Perform a local search over $\mathcal{H}_j$
    b) Identify the sides of the rectangle $\mathcal{H}_j$ and divide $\mathcal{H}_j$ into smaller hyper-rectangles along these directions
    c) Evaluate f at centers of new hyper-rectangles
    d) Update f_{max}, x_{max}, and m
  $k \leftarrow k + 1$ and update $\mathcal{P}_k, \mathcal{I}_k, \mathcal{C}_k, f_i(\forall i \in \mathcal{P}_k)$

Every hyper-rectangle i is represented by a pair (f_i, d_i), where f_i is the function value evaluated

at the centre of $\mathcal{H}_i$ and d_i is the size of the hyper-rectangle. The criterion to select hyper-rectangles, the so-called *potentially optimal hyper-rectangles*, for further divided is based on a score computed from (f_i, d_i). A pure local strategy would select the hyper-rectangle with the largest value for f_i, while a pure global search strategy would choose one of the hyper-rectangles with the biggest size d_i in each iteration. The main idea of the `DIRECT` algorithm is to balance between the local and global search, which can achieve by using a score weighting the two search strategies: $f_i + K \times d_i$ for some $K > 0$. The potentially optimal hyper-rectangles for `DIRECT` are defined as follows [53]:

Definition 5.1. *Let $\epsilon > 0$ be a small positive constant and f_{max} be the current best function value over Ω. We denote f_i by the function value at the centre of the hyper-rectangle $\mathcal{H}_i$. A hyper-rectangle j is said to be potentially optimal if there exists $K > 0$ such that*

$$f_j + K \times d_j \;\geq\; f_i + K \times d_i, \;\; \forall i \in [m] \tag{5.3}$$

$$f_j + K \times d_j \;\geq\; f_{max} + \epsilon, \tag{5.4}$$

where d_j is one half of the diameter of the hyper-rectangle $\mathcal{H}_j$.

The global convergence of Algorithm 5.1 is well established as `DIRECT` keeps splitting large hyper-rectangles into smaller ones. However, `DIRECT` is not sample-efficient because it needs to evaluate the target function for many times in each step. In addition, `DIRECT` only evaluates at the center of the hyper-rectangles which may not be optimal for searching better candidates.

5.3.2 Bayesian Optimization

Bayesian optimization is a sample-efficient global optimization method for solving (5.1). BO builds a surrogate function to approximate the unknown target function f using the Bayesian ideas.

At iteration t, given $D_t = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ where $y_i = f(x_i)$, the objective function is distributed according to a Gaussian process (GP) prior: $f(x) \sim GP(\mu(x), k(x, x'))$. For convenience, the mean function $\mu(x)$ is often assumed as a zero function. The covariance matrix $k(x, x')$ is used to model the covariance between any two function values $f(x)$ and $f(x')$. Here, we describe two examples that are widely used in practice.

- *Power exponential* kernel:

$$k(x, x') = \exp(-\frac{|x - x'|^2}{2\ell^2}).$$

- *Màtern* kernel:

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|d|}{\ell} \right)^{\nu} K_\nu(\frac{\sqrt{2\nu}|d|}{\ell})$$

  where $d = x - x'$, and K_ν is the modified Bessel function. One common selection for the parameter is $\nu = 5/2$, where the kernel function is reduced to

$$k(x, x') = (1 + \sqrt{5}|d| + \frac{5|d|^2}{3\ell}) \exp(-\frac{5|d|}{\ell}).$$

Under the GP prior and the data $\mathcal{D}_t$, the *predictive distribution* for $f(x)$ at any point x is again a Gaussian distribution with its mean

$$\mu_t(x) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{y}_{1:t} \tag{5.5}$$

and covariance

$$\sigma_t^2(x) = k(x, x) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}, \tag{5.6}$$

where $\mathbf{K} \in \mathbb{R}^{t \times t}$ has entries $k(x_i, x_j)$ and

$$\mathbf{k} = [k(x_1, x), \ldots, k(x_t, x)]^T.$$

In the sequential decision making setting, the next query point is chosen by optimizing the acquisition function. Typically, the acquisition function $\alpha(x|\mathcal{D}_t)$ is defined such that peak values would correspond to potentially high values of the objective function due to either high prediction or high uncertainty. We present two widely used acquisition functions in the BO setting:

- **Expected improvement (EI):** The expected improvement with respect to the best function value yet seen $f_t^* = \max\{f(x_1), \ldots, f(x_t)\}$ is defined by

$$\mathbf{EI}(x) = \mathbb{E}[f(x) - f_t^*]^+ \tag{5.7}$$

where $a^+ = \max(a, 0)$. Based on the GP posterior, one can easily compute such expectation, yielding:

$$\mathbf{EI}(x) = \sigma(x)(z\Phi(z) + \phi(z)) \tag{5.8}$$

where

$$z = \frac{\mu(x) - f_t^*}{\sigma(x)},$$

and $\phi(\cdot)$, $\Phi(\cdot)$ represent the pdf and cdf of the standard normal distribution respectively.

- **Upper confidence bound (UCB): UCB** considers the combination of mean and variance

$$\mathbf{UCB}(x) = \mu(x) + \kappa\sigma(x) \tag{5.9}$$

as the acquisition function.

5.4 Algorithm derivation

The `DIRECT` requires extensive function evaluations of the unknown target function $f(\mathbf{x})$. However, such requirement is unrealistic in the HPO setting as each function evaluation may take minutes even hours. This drawback prevents the practical application of `DIRECT` and motivates us to propose the new algorithm Bayesian DIRECT (`BD`) which can substantially reduce the function evaluations with the help of Bayesian models over the unknown target functions. The `BD` algorithm is described in Algorithm 5.2.

Algorithm 5.2 Bayesian DIRECT (`BD`)

Initialize with m points to get $\mathcal{P}_1, \mathcal{I}_1, \mathcal{D}_1$, and f_{max}; $k \leftarrow 1$
while $m \leq f_{eval}^{max}$ **do**
  Identify the set $\mathcal{S}$ of all potentially optimal hyper-rectangles in $\mathcal{P}_k$
  for $j \in \mathcal{S}$ **do**
    a) Find $\mathbf{x}_j = \text{argmax}_{\mathbf{x} \in \mathcal{H}_j} \alpha(\mathbf{x}|\mathcal{D}_k)$ and evaluate $f(\mathbf{x}_j)$
    b) Identify the sides of the rectangle $\mathcal{H}_j$ and divide $\mathcal{H}_j$ into smaller hyper-rectangles along these directions
    c) Update f_{max}, and m
  $k \leftarrow k + 1$ and update $\mathcal{P}_k, \mathcal{I}_k, \mathcal{D}_k$ and $\alpha(\mathbf{x}|\mathcal{D}_k)$

In the following subsections, we explain the details of the initialization step, and how to identify and split potentially optimal hyper-rectangles.

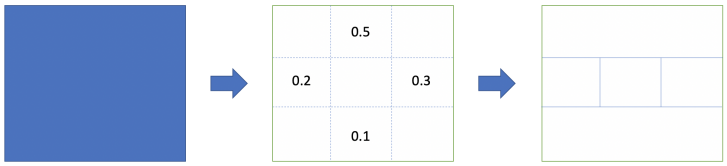


Figure 5.1: Initialization of BD

5.4.1 Initialization

In the initialization step, we randomly sample m points $\mathcal{D}_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset [0,1]^p$ using the Latin Hypercube sampling (LHS) algorithm and evaluate $f(\mathbf{x}_i), i \in [m]$. The posterior function $P(f(\mathbf{x})|\mathcal{D}_1)$ and the acquisition function $\alpha(\mathbf{x}|\mathcal{D}_1)$ are computed under the GP prior. Then, BD divides the hyper-cube Ω by maximizing the acquisition function in $2d$ hyper-cubes centering at $\mathbf{c}_1 + \delta\mathbf{e}_i$ and $\mathbf{c}_1 - \delta\mathbf{e}_i$ with $\mathbf{c}_1 = (0.5, \ldots, 0.5) \in \mathbb{R}^p$ and width $\delta = \frac{1}{3}$ and denote these hyper-cubes as $\Omega_{i,+}$ and $\Omega_{i,-}$ for $i \in [p]$. Let

$$\alpha_{i,+} = \underset{\mathbf{x}\in\Omega_{i,+}}{\operatorname{argmax}}\, \alpha(\mathbf{x}|\mathcal{D}) \quad \text{and} \quad \alpha_{i,-} = \underset{\mathbf{x}\in\Omega_{i,-}}{\operatorname{argmax}}\, \alpha(\mathbf{x}|\mathcal{D})$$

for $i \in [p]$. Following the idea of DIRECT, BD select a hyper-rectangle with a large acquisition function value in the search space; hence let us define

$$\alpha_i = \max\{\alpha_{i,+}, \alpha_{i,-}\}, i \in [p]$$

and the dimension with largest α is partitioned into thirds. Once this is done, split the hyper-rectangle into thirds along the dimension with next largest α value. Continue in this way until we have split on all dimensions. By doing so, $\mathbf{c}_1 \pm \delta\mathbf{e}_i$ are the center of the newly generated hyper-rectangles and we can initialize the sets $\mathcal{P}_1, \mathcal{I}_1$ simultaneously. Figure 5.1 shows the way of splitting the hyper-rectangle for the case $p = 2$.

5.4.2 Potentially Optimal Hyper-rectangles

In each iteration, we use a new criteria for BD to select the next potentially optimal hyper-rectangles which should be divided. Suppose that we use the **UCB** as the acquisition function, BD

searches locally and globally by dividing all hyper-rectangles that meet the criteria in Definition 5.2.

Definition 5.2. *Let α_i is the best acquisition function value over the hyper-rectangle $\mathcal{H}_i$. A hyper-rectangle j is said to be potentially optimal if there exists $K \geq 0$ such that*

$$\alpha_j + K \times d_j \;\geq\; \alpha_i + K \times d_i, \quad \textit{for all } i \in [m] \tag{5.10}$$

$$\alpha_j + K \times d_j \;\geq\; f_{max} \tag{5.11}$$

where d_j represents the center-vertex distance of the hyper-rectangle j.

Here, we use the definition for d_j as the same one used in [52]. Instead of using the exact function value in each hyper-rectangle, we replace f_j by the best acquisition function value α_j over the hyper-rectangles. By doing so, `BD` provides a sample-efficient approach to detect the potentially optimal hyper-rectangle for the next iteration.

Lemma 5.1. *Let $\mathcal{I}$ be the set of all indices of all existing hyper-rectangles and for each $j \in \mathcal{I}$, define*

$$\mathcal{I}_1 = \{i \in \mathcal{I} : d_i < d_j\}$$

$$\mathcal{I}_2 = \{i \in \mathcal{I} : d_i > d_j\}$$

$$\mathcal{I}_3 = \{i \in \mathcal{I} : d_i = d_j\}$$

and

$$g_i = \frac{\alpha_j - \alpha_i}{d_i - d_j}, \quad \forall i \in \mathcal{I}_1 \cup \mathcal{I}_2.$$

If the hyper-rectangle j is potentially optimal, then

$$\alpha_j \geq \alpha_i, \quad \forall i \in \mathcal{I}_3, \tag{5.12}$$

there exists $K > 0$ such that

$$\max_{i \in \mathcal{I}_1} g_i \leq K \leq \min_{i \in \mathcal{I}_2} g_i, \tag{5.13}$$

and

$$\alpha_j + \min_{i \in \mathcal{I}_2} g_i \times d_j \geq f_{max}. \tag{5.14}$$

Proof. For $i \in \mathcal{I}_3$, the inequality $\alpha_j \geq \alpha_i$ follows directly from (5.10). For $i \in \mathcal{I}_1$, we have

$$K \geq \frac{\alpha_j - \alpha_i}{d_i - d_j},$$

and for $i \in \mathcal{I}_2$, it implies that

$$K \leq \frac{\alpha_j - \alpha_i}{d_i - d_j}.$$

Hence, (5.13) directly follows from above by taking the maximum over $\mathcal{I}_1$ and taking the minimum over $\mathcal{I}_2$. If $\min_{i \in \mathcal{I}_2} g_i \geq \max\{0, \max_{i \in \mathcal{I}_1} g_i\}$, taking the maximum for K gives the result in (5.14). □

The following figure illustrates the selection process given the pairs (α_i, d_i), $\forall i \in \mathcal{I}$. Each point represents a hyper-rectangle in the current iteration.

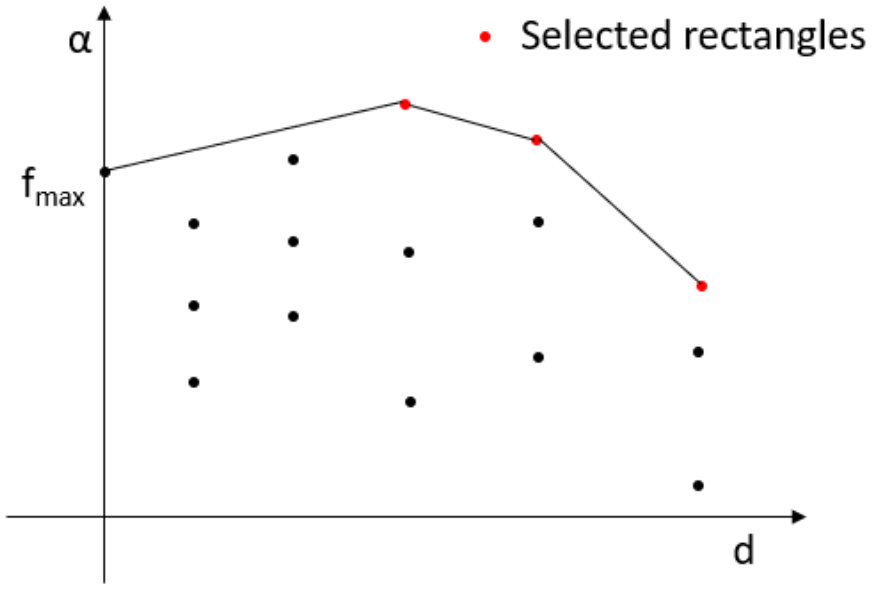


Figure 5.2: Identifying potentially optimal hyper-rectangles for `BD`

If we use the **EI** as the acquisition function in `BD`, the potentially optimal hyper-rectangle is defined in the following statement.

Definition 5.3. *Let $\epsilon > 0$ be a small positive constant and α_i is the best acquisition function value over the hyper-rectangle $\mathcal{H}_i$. A hyper-rectangle j is said to be potentially optimal if there exists $K \geq 0$ such that*

$$\alpha_j + K \times d_j \geq \alpha_i + K \times d_i, \quad \textit{for all } i \in [m] \tag{5.15}$$

$$\alpha_j + K \times d_j \geq \epsilon |f_{max} - f_{median}| \tag{5.16}$$

where d_j represents the center-vertex distance of the hyper-rectangle j.

The small positive constant ϵ is used to balance between global search (exploration) and local search (exploitation). A common choice for ϵ is 0.001 and we will use it as the default value when using **EI** acquisition function in `BD`. f_{median} is the median of all previous evaluated target function values in `BD`.

5.4.3 Splitting Potentially Optimal Hyper-rectangles

Once a hyper-rectangle has been identified potentially optimal, `BD` divides this hyper-rectangle into smaller hyper-rectangles. The algorithm only splits the dimension with the longest width. In k-th iteration, for any potentially optimal hyper-rectangle $\mathcal{H} \in \mathcal{P}_k$ with center $\mathbf{c}$, let $J \subset [p]$ be the index set such that the j-th dimension has the longest width w for any $j \in J$. Following the same idea in the initialization step, `BD` divides the selected hyper-rectangle $\mathcal{H}$ by maximizing the acquisition function in $2|J|$ hyper-cubes centering at $\mathbf{c} + \frac{w}{3} e_j$ and $\mathbf{c} - \frac{w}{3} e_j$ with width $\frac{w}{3}$ for $j \in J$. The dimension with largest acquisition function value will be selected and is partitioned into thirds. Continue in this way until all dimensions in J are split. Figure 5.3 gives an example of how to split the Potentially Optimal Hyper-rectangles in $\mathbb{R}^2$.

Figure 5.3: Splitting Hyper-rectangles in BD

5.5 Convergence results

In this section, we analyze the theoretical convergence of BD using the **UCB** acquisition function. The following theorem proves that the sequence generated by the BD is dense in the feasible region and the global convergence of BD follows directly from such observation as long as the target function is continuous.

Theorem 5.1. *The set $\cup_{k=1}^{\infty}\mathcal{D}_k$ generated by Algorithm 5.2 is dense in Ω.*

Proof. In iteration k of BD, let $\mathcal{I}_k^{max} \subset \mathcal{I}_k$ be the index set for the hyper-rectangles with the largest value d. Let $i_k \in \mathcal{I}_k^{max}$ be the index such that

$$\alpha_{i_k} \geq \alpha_j, \forall j \in \mathcal{I}_k^{max}.$$

From the definition of 5.2, the hyper-rectangle $\mathcal{H}_{i_k}$ will be selected as potentially optimal and a new point $\mathbf{x}_{i_k}^* \in \mathcal{H}_{i_k}$ will be evaluated. Based on this observation, we can conclude at least one of the hyper-rectangles with largest value d will be split in each iteration. Thus, the points generated by BD are dense in Ω. □

The next Lemma claims BD will select the point with the largest acquisition function value α which coincides with the BO algorithm.

Lemma 5.2. *In each iteration, the hyper-rectangle with the largest acquisition function value α will be selected as a potentially optimal hyper-rectangle.*

Proof. In iteration k, let $\mathcal{I}_k$ be the set of all indices of all existing hyper-rectangles and

$$\mathbf{x}_i^* = \underset{\mathbf{x}\in\mathcal{H}_i}{\operatorname{argmax}}\, \alpha(\mathbf{x}|\mathcal{D}_k) \text{ and } \alpha_i = \alpha(\mathbf{x}_i^*|\mathcal{D}_k), \quad \forall i = 1, \ldots, |\mathcal{I}_k|.$$

Let

$$i_k = \underset{i \in \mathcal{I}_k}{\operatorname{argmax}}\, \alpha_i$$

and it follows directly that

$$\mathbf{x}_{i_k} = \underset{\mathbf{x} \in \Omega}{\operatorname{argmax}}\, \alpha(\mathbf{x}|\mathcal{D}_k).$$

Recall the definition of **UCB** acquisition function that $\alpha(\mathbf{x}|\mathcal{D}_k) = \mu_k(\mathbf{x}) + \kappa\sigma_k(\mathbf{x})$. Then

$$\alpha_{i_k} = \max_{\mathbf{x} \in \Omega} \alpha(\mathbf{x}|\mathcal{D}_k) > \alpha(\mathbf{x}_{max}|\mathcal{D}_k) = \underbrace{\mu_k(\mathbf{x}_{max})}_{f_{max}} + \underbrace{\kappa\sigma_k(\mathbf{x}_{max})}_{0} = f_{max}.$$

As a result, (5.11) holds for any $K \geq 0$. Since $\alpha_{i_k} = \max_{i \in \mathcal{I}_k} \alpha_i$. There exists a sufficient small $\epsilon_0 > 0$ such that for any $0 \leq K \leq \epsilon_0$

$$\alpha_{i_k} + K \times d_{i_k} \geq \alpha_i + K \times d_i, \quad \forall i \in \mathcal{I}_k.$$

Combining the above two observations, we can conclude that the hyper-rectangle i_k will be selected as potentially optimal. □

We have proved that `BD` will select at least two new points (they might overlap) in each iteration. However, it doesn't mean only these two points will be included and there might be some other hyper-rectangles selected by `BD` as long as they satisfy the requirements by Definition 5.2. Figure 5.2 gives a good visualization of the selection process.

5.6 Numerical examples

We evaluate the performance of the proposed algorithm on several synthetic benchmarks. Moreover, we examine its ability to tune the hyper-parameters for the random forest, logistic regression, and deep learning on some well-known datasets.

5.6.1 Synthetic Test Functions

We evaluate `BD` and `BO` on five test functions chosen from [94], including the 2D *Branin* function on $[-5, 15] \times [0, 15]$, 3D *Hartmann* function on $[0, 1]^3$, 3D *Rosenbrock* function on $[-2, 2]^3$, 4D *Levy*

function on $[-10, 10]^4$, and 5D *Ackley* on $[-2, 2]^5$. For the acquisition function, we test both **EI** and **UCB** on `BD` and `BO`. We run the algorithms for 50 trials and report the average of best observed target function values in Figure 5.4. `BO` often outperforms in the early 20-40 function evaluations while `BD` shows a better convergence in later updates. This phenomenon is in line with our expectation as `BD` needs more function evaluations in the early iterations to discover potentially optimal regions while `BO` makes greedy progress from the beginning.

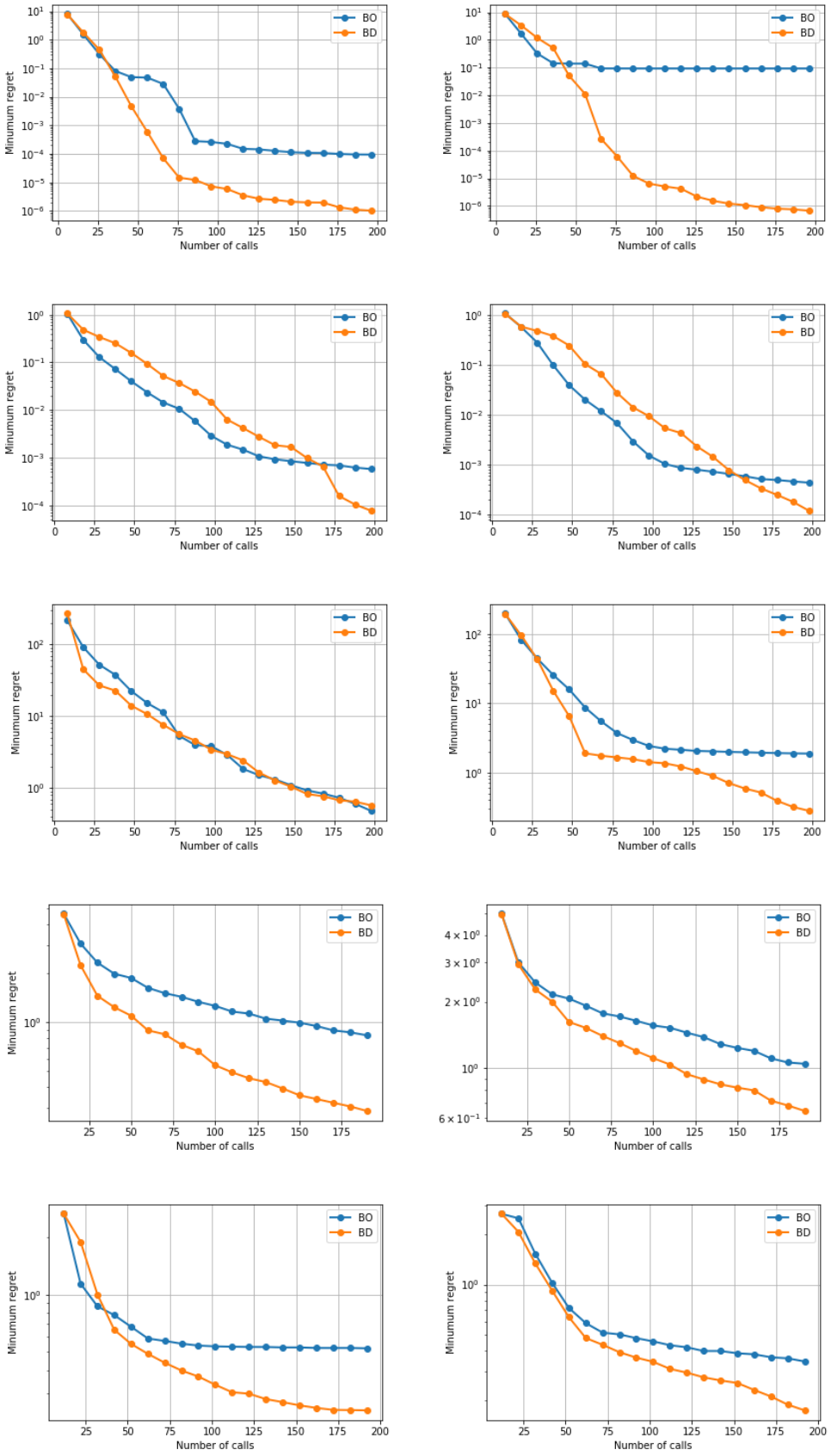


Figure 5.4: **EI** and **UCB** (from left to right). *Branin*, *Hartmann*, *Rosenbrock*, *Levy*, and *Ackley* (from top to bottom)

5.6.2 Random Forest for Binary Classification

Random Forest [12] is a tree ensemble method which utilizes the bagging strategy. We run a random forest classifier on four benchmark binary classification datasets [1], including *a1a*, *mushrooms*, *svmguide3*, and *w1a*. We tune 3 hyperparameters for random forest: *n_estimators*, *min_samples_split*, and *max_features*. The *n_estimators* determines the number of trees in the classifier and is selected from 10 to 250. The *min_samples_split* represents the minimum number of samples required to split an internal node and takes values between 3 and 25. The *max_features* chooses the percentage of features from 0.01 to 0.99 to consider when looking for the best split. The output is defined as the cross-validation score with negative logistic loss. Figure 5.5 reports the average for 20 times run of both algorithms.

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html

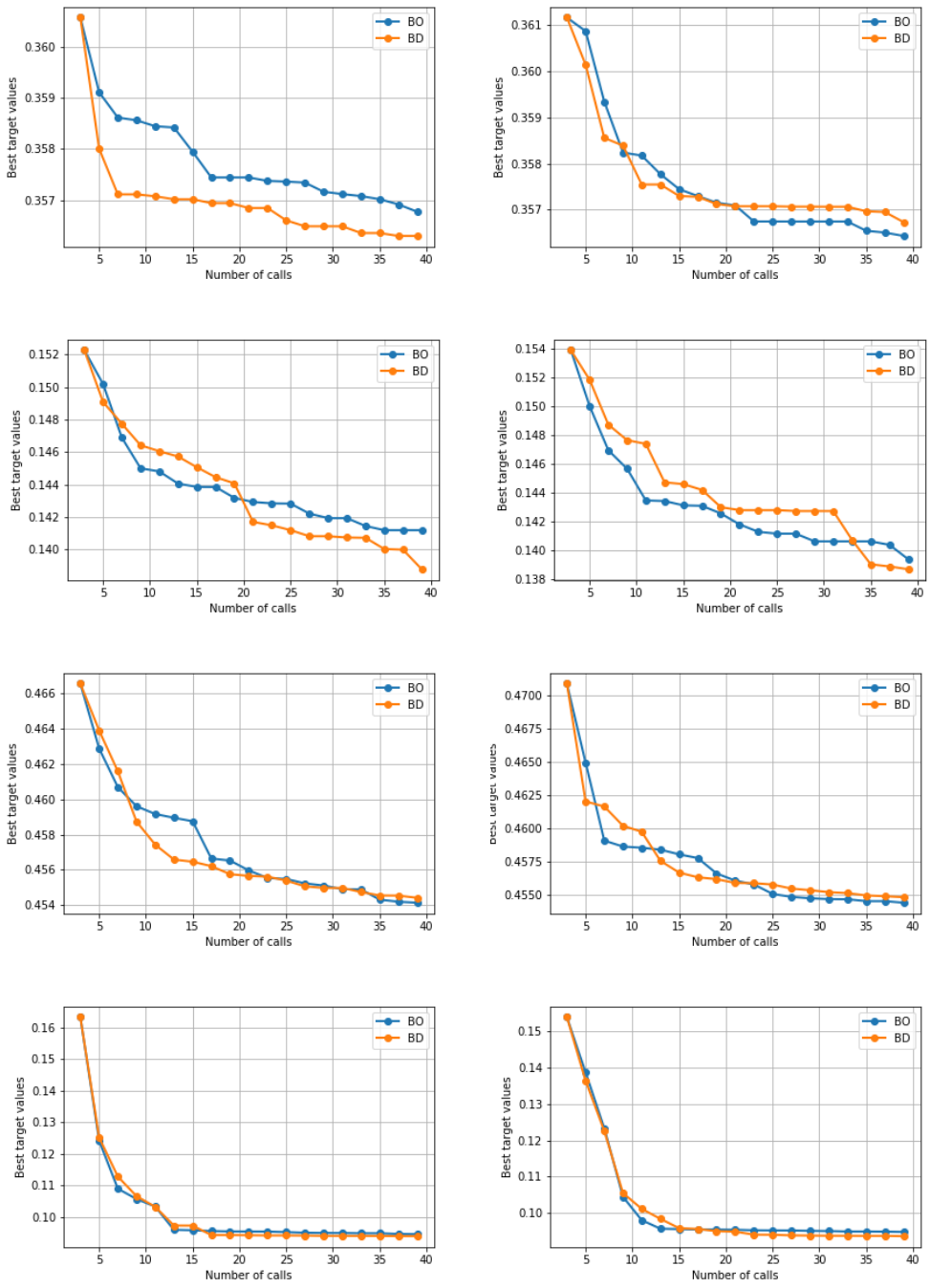


Figure 5.5: **EI** and **UCB** (from left to right). *a1a*, *mushrooms*, *svmguide3*, and *w1a* datasets (from top to bottom).

5.6.3 Logistic Regression and Deep Learning

We tune logistic regression and a feedforward neural network with 2 hidden layers on the MNIST dataset, a standard classification task for handwritten digits. The training set contains

60000 images, and the test set 10000. We tune 4 hyperparameters for logistic regression: the ℓ_2 regularization parameter from $1e-6$ to 1, learning rate from $1e-6$ to 1, mini batch size from 10 to 1000 and training epochs from 5 to 20. For the neural network, we additionally tune the number of hidden units in $[100, 1000]$. For all of the 5 hyperparameters, the training epoch uses the regular scale while the other 4 parameters are selected in the $\log_{10}$ scale. Figure 5.6 reports the average for 20 times run of both algorithms.

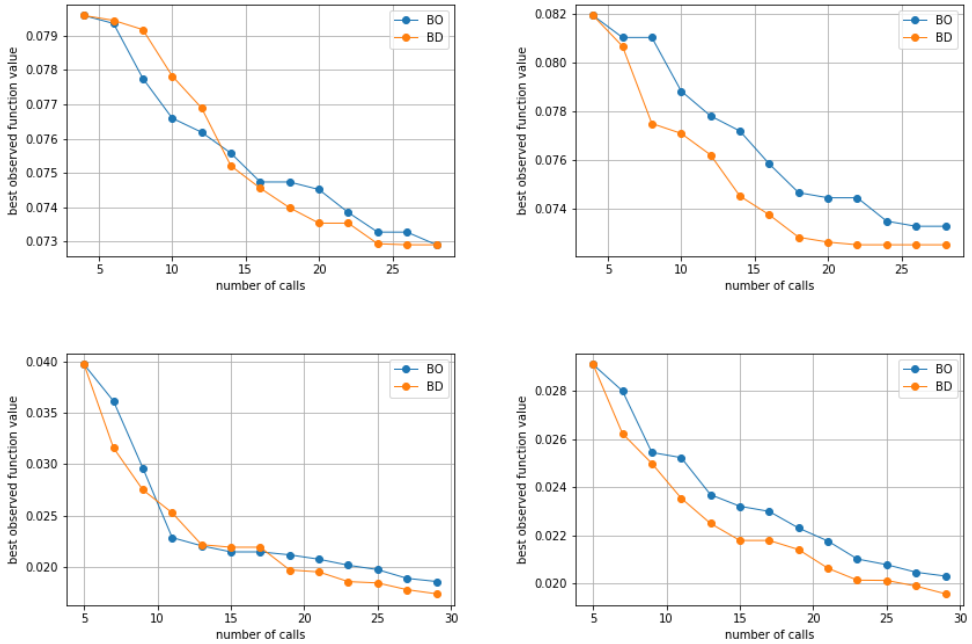


Figure 5.6: **EI** and **UCB** (from left to right). Logistic regression and deep learning (from top to bottom).

REFERENCES

[1] Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18:1–40, 2016.

[2] Zeyuan Allen-Zhu and Yang Yuan. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *International conference on machine learning*, pages 1080–1089, 2016.

[3] Galen Andrew and Jianfeng Gao. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[4] Charles Audet and John E Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006.

[5] Jianchao Bai, Hongchao Zhang, and Jicheng Li. A parameterized proximal point algorithm for separable convex optimization. *Optimization Letters*, pages 1–20, 2017.

[6] Amir Beck, Angelia Nedic, Asuman Ozdaglar, and Marc Teboulle. An $O(1/k)$ Gradient Method for Network Resource Allocation Problems. *IEEE Transactions on Control of Network Systems*, 1(1):64–73, 2014.

[7] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[8] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2018.

[9] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[10] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[12] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[14] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[15] Tsung-Hui Chang, Angelia Nedic, and Anna Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automatic Control*, 59(6):1524–1538, 2014.

[16] Nikolaos Chatzipanagiotis, Darinka Dentcheva, and Michael M Zavlanos. An augmented Lagrangian method for distributed optimization. *Mathematical Programming*, 152(1-2):405–434, 2015.

[17] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

[18] Gong Chen and Marc Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathematical Programming*, 64(1-3):81–101, 1994.

[19] Shaobing Chen and David Donoho. Basis pursuit. In *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44. IEEE, 1994.

[20] Alberto Costa and Giacomo Nannicini. Rbfopt: an open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation*, 10(4):597–629, 2018.

[21] Ying Cui, Defeng Sun, and Kim-Chuan Toh. On the R-superlinear convergence of the KKT residues generated by the augmented Lagrangian method for convex composite conic programming. *arXiv preprint arXiv:1706.08800*, 2017.

[22] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[23] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel Multi-Block ADMM with o(1/k) Convergence. *Journal of scientific computing*, 71(2):712–736, 2017.

[24] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.

[25] G. Di Pillo, G. Liuzzi, S. Lucidi, V. Piccialli, and F. Rinaldi. A DIRECT-type approach for derivative-free constrained global optimization. *Computational Optimization and Applications*, 65(2):361–397, Nov 2016.

[26] Asen L Dontchev. Implicit functions and solution mappings. *Springer Monographs in Mathematics*, 2009.

[27] Jonathan Eckstein and Dimitri P Bertsekas. On the Douglasa–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

[28] Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *International Conference on Neural Information Processing Systems*, pages 3052–3060, 2015.

[29] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[30] D. E. Finkel and C. T. Kelley. Additive scaling and the DIRECT algorithm. *Journal of Global Optimization*, 36(4):597–608, Dec 2006.

[31] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[32] Jorg Maximilian Xaver Gablonsky. *Modifications of the Direct Algorithm*. PhD thesis, North Carolina State University, Raleigh, North Carolina, 2001. AAI3030042.

[33] Ratko Grbić, Emmanuel Karlo Nyarko, and Rudolf Scitovski. A modification of the DIRECT method for lipschitz global optimization for a symmetric function. *Journal of Global Optimization*, 57(4):1193–1212, Dec 2013.

[34] Osman Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[35] Deren Han, Defeng Sun, and Liwei Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite quadratic and semi-definite programming. *arXiv preprint arXiv:1508.02134*, 2015.

[36] Deren Han, Defeng Sun, and Liwei Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Mathematics of Operations Research*, 43(2):622–637, 2017.

[37] Deren Han and Xiaoming Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155(1):227–238, 2012.

[38] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[39] Bingsheng He, Li-Zhi Liao, Deren Han, and Hai Yang. A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming*, 92(1):103–118, 2002.

[40] Bingsheng He, Min Tao, and Xiaoming Yuan. Alternating direction method with gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2):313–340, 2012.

[41] Bingsheng He, Min Tao, and Xiaoming Yuan. Convergence rate analysis for the alternating direction method of multipliers with a substitution procedure for separable convex programming. *Mathematics of Operations Research*, 42(3):662–691, 2017.

[42] Bingsheng He and Xiaoming Yuan. On the acceleration of augmented lagrangian method for linearly constrained optimization. *Optimization online*, 3, 2010.

[43] Bingsheng He and Xiaoming Yuan. On the O(1/n) Convergence Rate of the Douglasa–Rachford Alternating Direction Method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[44] Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.

[45] Alan J Hoffman. On approximate solutions of systems of linear inequalities. *Selected Papers Of Alan J Hoffman: With Commentary*, pages 174–176, 2003.

[46] John T Holodnak and Ilse CF Ipsen. Randomized approximation of the gram matrix: Exact computation and probabilistic bounds. *SIAM Journal on Matrix Analysis and Applications*, 36(1):110–137, 2015.

[47] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

[48] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[49] Robert Hooke and Terry A Jeeves. "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

[50] Waltraud Huyer and Arnold Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, Jun 1999.

[51] Jinzhu Jia, Fang Xie, and Lihu Xu. Sparse poisson regression with penalized weighted score function. *arXiv preprint arXiv:1703.03965*, 2017.

[52] Donald R. Jones. *Direct global optimization algorithmDirect Global Optimization Algorithm*, pages 431–440. Springer US, Boston, MA, 2001.

[53] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79(1):157–181, 1993.

[54] Xudong Li, Defeng Sun, and Kim-Chuan Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. *arXiv preprint arXiv:1607.05428*, 2016.

[55] Hongsheng Liu and Shu Lu. Convergence of the augmented decomposition algorithm. *Computational Optimization and Applications*, 72(1):179–213, 2019.

[56] Qinghua Liu, Xinyue Shen, and Yuantao Gu. Linearized admm for nonconvex nonsmooth optimization with convergence analysis. *IEEE Access*, 2019.

[57] Qunfeng Liu. Linear scaling and the direct algorithm. *Journal of Global Optimization*, 56(3):1233–1245, 2013.

[58] Yong-Jin Liu, Defeng Sun, and Kim-Chuan Toh. An implementable proximal point algorithmic framework for nuclear norm minimization. *Mathematical programming*, 133(1):399–436, 2012.

[59] G. Liuzzi, S. Lucidi, and V. Piccialli. Exploiting derivative-free local searches in DIRECT-type algorithms for global optimization. *Computational Optimization and Applications*, 65(2):449–475, Nov 2016.

[60] G. Liuzzi, Stefano Lucidi, and Veronica Piccialli. A direct-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Computational Optimization and Applications*, 45:353–375, 03 2010.

[61] Giampaolo Liuzzi, Stefano Lucidi, and Veronica Piccialli. A partition-based global optimization algorithm. *Journal of Global Optimization*, 48:113128, 2010.

[62] Zhaosong Lu. Randomized block proximal damped newton method for composite self-concordant minimization. *SIAM Journal on Optimization*, 27(3):1910–1942, 2017.

[63] Zhi-Quan Luo and Paul Tseng. On the convergence rate of dual ascent methods for linearly constrained convex minimization. *Mathematics of Operations Research*, 18(4):846–867, 1993.

[64] Fernando Javier Luque. Asymptotic convergence analysis of the proximal point algorithm. *SIAM Journal on Control and Optimization*, 22(2):277–293, 1984.

[65] Shiqian Ma. Alternating proximal gradient method for convex minimization. *Journal of Scientific Computing*, 68(2):546–572, 2016.

[66] James Stephen Marron, Michael J Todd, and Jeongyoun Ahn. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271, 2007.

[67] Jefferson G Melo and Renato DC Monteiro. Iteration-complexity of a linearized proximal multiblock admm class for linearly constrained nonconvex optimization problems. *http://www. optimization-online. org*, 2017.

[68] Jonas Mockus. *Bayesian approach to global optimization: theory and applications*, volume 37. Springer Science & Business Media, 2012.

[69] John M Mulvey, Andrzej Ruszczyn, et al. A diagonal quadratic approximation method for large scale linear programs. *Operations Research Letters*, 12(4):205–215, 1992.

[70] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[71] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.

[72] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[73] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621. JMLR. org, 2017.

[74] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[75] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2233–2246, 2012.

[76] Mert Pilanci and Martin J. Wainwright. Newton sketch: A linear-time optimization algorithm with linear-quadratic convergence. *Mathematics*, 27(1), 2015.

[77] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[78] Stephen M Robinson. Some continuity properties of polyhedral multifunctions. *Mathematical Programming at Oberwolfach*, pages 206–214, 1981.

[79] R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1970.

[80] R Tyrrell Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976.

[81] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[82] R Tyrrell Rockafellar. PROBLEM DECOMPOSITION IN BLOCK-SEPARABLE CONVEX OPTIMIZATION: IDEAS OLD AND NEW. *Washington.edu*, 2017.

[83] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods i: globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016.

[84] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods ii: Local convergence rates. *arXiv preprint arXiv:1601.04738*, 2016.

[85] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[86] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie*, volume 1. Springer, 1977.

[87] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

[88] Ron Shefi and Marc Teboulle. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, 24(1):269–297, 2014.

[89] Jonathan E Spingarn. Applications of the method of partial inverses to convex programming: decomposition. *Mathematical Programming*, 32(2):199–223, 1985.

[90] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. Mit Press, 2012.

[91] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[92] Kaizhao Sun and X Andy Sun. A two-level distributed algorithm for general constrained non-convex optimization with global convergence. *arXiv preprint arXiv:1902.07654*, 2019.

[93] Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: A recipe for newton-type methods. *arXiv preprint arXiv:1703.04599*, 2017.

[94] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved April 23, 2020, from http://www.sfu.ca/~ssurjano.

[95] Min Tao and Xiaoming Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[96] Arash Tavassoli, Kambiz Haji Hajikolaei, Soheil Sadeqi, G. Gary Wang, and Erik Kjeang. Modification of direct for high-dimensional design problems. *Engineering Optimization*, 46(6):810–823, 2014.

[97] Virginia Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, 1991.

[98] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1–25, 1997.

[99] Quoc Tran-Dinh, Anastasios Kyrillidis, and Volkan Cevher. Composite self-concordant minimization. *The Journal of Machine Learning Research*, 16(1):371–416, 2015.

[100] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[101] Paul Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138, 1991.

[102] Emmanuel Vazquez and Julien Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095, 2010.

[103] Xiangfeng Wang, Mingyi Hong, Shiqian Ma, and Zhi-Quan Luo. Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers. *arXiv preprint arXiv:1308.5294*, 2013.

[104] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

[105] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3627–3635. JMLR. org, 2017.

[106] Stephen J Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186, 2012.

[107] Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pages 5267–5278, 2017.

[108] Lin Xiao and Stephen Boyd. Optimal scaling of a gradient method for distributed resource allocation. *Journal of optimization theory and applications*, 129(3):469–488, 2006.

[109] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[110] Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

[111] Haishan Ye, Luo Luo, and Zhihua Zhang. Approximate newton methods and their local convergence. In *International Conference on Machine Learning*, pages 3931–3939, 2017.

[112] Keyou You and Lihua Xie. Network topology and communication data rate for consensusability of discrete-time multi-agent systems. *IEEE Transactions on Automatic Control*, 56(10):2262–2275, 2011.